AD/

Annual Report

Defense Switched Network Technology and Experiments Program

30 September 1985

Lincoln Laboratory

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

LEXINGTON, MASSACHUSETTS

ADA169741

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LINCOLN LABORATORY

DEFENSE SWITCHED NETWORK TECHNOLOGY
AND EXPERIMENTS PROGRAM

ANNUAL REPORT
TO THE
DEFENSE COMMUNICATIONS AGENCY

1 OCTOBER 1984 — 30 SEPTEMBER 1985

ISSUED 22 MAY 1986

LEXINGTON   MASSACHUSETTS

ABSTRACT

This report documents work performed during FY85 on the DCA-sponsored Defense Switched Network Technology and Experiments Program. The areas of work reported are: (1) development and evaluation of routing and system control techniques potentially applicable in the Defense Switched Network (DSN), including investigation of Machine Intelligence techniques for system control; (2) instrumentation and integration of the Experimental Integrated Switched Network (EISN) test-bed facility; and (3) EISN experiment planning and system coordination.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

DEFENSE SWITCHED NETWORK TECHNOLOGY AND EXPERIMENTS PROGRAM

1. INTRODUCTION AND SUMMARY

This report documents work performed during FY85 on the DCA-sponsored Defense Switched Network Technology and Experiments Program. The areas of work reported are: (a) development and evaluation of routing and system control techniques potentially applicable in the Defense Switched Network (DSN), including investigation of Machine Intelligence techniques for system control; (b) instrumentation and integration of the Experimental Integrated Switched Network (EISN) test-bed facility; and (c) EISN experiment planning and system coordination.

FY85 has been the final year of the Defense Switched Network Technology and Experiments Program, concluding a series of DCA-sponsored programs at Lincoln Laboratory that has evolved over some ten years. These programs have included: Speech Evaluation (development and test of narrowband speech coding systems), 1975-76; Network Speech System Implications of Packet Speech, 1976; Network Speech Processing, 1977-78; Voice Conferencing Technology, 1977-79; Network Speech Systems Technology, 1979-81; and Defense Switched Network Technology and Experiments, 1982-84. A bibliography of Lincoln publications on all of these programs is included at the end of this report. Work conducted during FY85 was directed at completion of the objectives that have guided efforts in the DSN Technology and Experiments Program for the last several years.[1-3] To permit orderly completion of reports and remaining details, the program was given a three-month no-cost extension so that the formal termination date of the program was 31 December 1985.

The design and implementation of the outboard Routing/Control Processor (RCP), controlling a modern off-the-shelf digital switch via an operator console interface, was completed.[4,5] The EISN Advanced Routing and System Control Test Bed was geographically deployed at two Army sites and in-house at Lincoln, as described in Section 3 of this report, and facilities were implemented for orderly and convenient setup and execution of experiments. Numerous series of real and phantom calling experiments were carried out, both to validate test-bed operation and to substantiate analytic and simulation results on routing and preemption performance, as described in detail in Section 2.2.

A major milestone, completed in FY85, was the integration and installation of a 3-node EISN facility at RADC. This installation represents a transfer of the EISN technology to RADC, and will be used as a major subsystem of a larger telecommunications test bed to be instrumented at RADC. In addition, plans call for the EISN facility at RADC to be used for future experiments in knowledge-based systems analysis and control, as noted below and described further in Section 2.5.

The Call-by-Call Simulator was completed,[6-8] and was applied in sweeping comparative evaluations of the new routing and preemption algorithms developed at Lincoln over the past several years, relative to the performance of standard techniques in current use in civilian and military telecommunications. These results are presented in Sections 2.3 and 2.4.

1

An important ongoing Lincoln role throughout the EISN program has been that of system-wide planner and coordinator. There have been four main elements in this activity: (a) system engineering for the Wideband Satellite Network, (b) developing the EISN test-bed facilities, (c) planning and coordinating the installation of these facilities throughout the network, and (d) planning and coordinating experiments on the EISN test bed. Substantial progress has been made in these areas in FY85, from achieving substantial improvement in satellite network operation through completion of a wide range of system experiments, including a set of experiments planned and executed by Army site personnel without Lincoln intervention. These activities are described in Section 4.

Looking to the future, a study was conducted on the feasibility of applying the knowledge gained in EISN to automation of the worldwide System Control of the Defense Communications System, taking advantage of the state of the art in Machine Intelligence. The study effort, and the conclusions reached, are described in Section 2.5. Operating military communications control centers were visited in the course of the study, and skilled personnel were interviewed and consulted as to desirable objectives and approaches for automation. Interactions took place with personnel at Rome Air Development Center who are pursuing the future development of System Control, and in fact RADC has undertaken to sponsor an FY86 program at Lincoln Laboratory to address the recommendations resulting from the FY85 study.

2. DSN ROUTING AND SYSTEM CONTROL TECHNIQUES EVALUATION

2.1 INTRODUCTION

Lincoln efforts in this program in previous years[1-3] have led to the development of new routing and preemption techniques directed at the dual DSN requirements of survivability and low cost. Encouraging early results had been obtained through the use of steady-state network design and analysis programs originally obtained from DCEC and modified to accommodate such new techniques as treating broadcast satellite channels differently from land lines. The new procedures were refined and extended in subsequent years, emphasizing the capability to respond effectively to major network damage. It was realized that the available analysis tools were unable to provide information about the dynamics of the new procedures in the critical periods after the occurrence of damage, and hence a Call-by-Call Simulator[7-9] was implemented, capable of simulating every call in an AUTOVON-size network. Successive versions of the Call-by-Call Simulator have been extensively used in the Lincoln Program, and have also been delivered to DCEC for use in other programs.

Concurrently, Lincoln developed a distributed hardware/software EISN Advanced Routing and System Control Test Bed[4,5] featuring modern off-the-shelf telephone switches with programmable outboard processors, aimed at providing independent confirmation of the performance of the new algorithms and of the Call-by-Call Simulator itself, as well as solving many of the problems that must be faced in creating a physical implementation of the new algorithms and procedures.

Both the Call-by-Call Simulator and the distributed test bed reached completion during FY85, and were used to obtain an extensive body of comparative results on the performance of the new techniques. Section 2.2 describes the experiments that were performed with the distributed EISN Test Bed, including real-call experiments which validated the implementations of the Outboard Processor and the Common-Channel Signaling (CCS) protocols (Subsection 2.2.1), as well as phantom call experiments employing independent emulated traffic generators in the distributed switch processors (Subsection 2.2.2). Section 2.3 describes the completion of the final features and capabilities to be added to the Call-by-Call Simulator, and Section 2.4 describes an extensive series of simulation runs performed with it to compare all the routing procedures under investigation, for normal conditions and for stressed conditions after various levels of damage, under a variety of offered traffic models. Section 2.4 includes a detailed discussion of the results, showing substantial improvement relative to the routing and preemption procedures in current use. Section 2.5 describes the Machine Intelligence study, together with the results and recommendations derived from it.

3

2.2 MULTISITE EISN EXPERIMENTS

In FY85 it became possible, with the completion and deployment of multiple RCPs, to conduct distributed routing experiments to substantiate earlier analysis and simulation results. Two complementary classes of experiments were carried out under various conditions, namely, real- and phantom-call experiments. The former consisted of RCP-controlled setup of actual voice circuits between human callers at physically separated EISN nodes, and the latter consisted of RCP-controlled setup of phantom circuits between traffic generators or precalculated call files at the nodes. Real-call experiments explore call setup performance as perceived by the user, including setup and transmission delay as well as satellite and terrestrial voice quality. Phantom-call experiments exercise the same RCP processing and CCS message exchange facilities as the real calls, but do not need to establish end-to-end voice circuits; call initiation rates and durations are determined by random processes which can be run at a much higher rate than would be feasible for real calls on the limited number of EISN telephones, thus permitting testing under realistic traffic load and system stress levels. The following sections report the results of a variety of experiments in both of these categories.

2.2.1 Real-Call Experiments

It had originally been planned that five RCPs would be implemented and installed at Lincoln Laboratory and exploited there for an extended series of real- and phantom-call experiments, before commencing the transfer of RCPs to DCEC and the military sites. As noted in Section 3 below, however, it became apparent early in FY85 that it would be desirable to speed up RCP delivery to the Army sites to permit the maximum amount of experimentation before the 30 September termination of Army EISN funding. The result was a capability to conduct multisite RCP experiments between UTX-1200 switches at Lincoln and Ft. Huachuca by February 1985, with the Ft. Monmouth SL-1 added to the net by March.

The first significant multisite calling experiments occurred on 21 February between Lincoln and Ft. Huachuca, using a long-distance modem call to carry the CCS traffic. No actual voice trunks were involved with these initial experiments, so they were in fact phantom calls, although the objective was to demonstrate multisite calling rather than to evaluate processing limits or obtain comparative statistics. After correction of some installation and start-up problems (as noted below), the Ft. Monmouth RCP facilities became available for multisite experimentation. In the mid-March time frame a series of RCP-controlled real-call experiments (using dialed-up terrestrial trunks) was conducted along each leg of the triangle formed by Lincoln and the two Army sites.

Since the RCPs and interface cabinets were in effect being shipped out to the Army sites as fast as they could be assembled and integrated, Lincoln had only a single in-house node (the first RCP, interfaced with one of the UTX-1200 switches) until late March. At that time the second UTX-1200 at Lincoln was integrated with its RCP, and in-house two-node system operation was demonstrated by means of a series of real calls using CCS and voice lines dialed up through both the Lincoln PBX and the Lexington Central Office.

4

A key Lincoln emphasis throughout the program has been to encourage an independent capability for the military site personnel to plan and execute EISN experiments tailored to their own objectives. A very positive result of this policy was the successful completion of a series of real-call experiments in April between the two Army sites, planned and executed by site personnel with no Lincoln involvement.

A benchmark RCP capability that was achieved in June was multilevel preemption on inter-switch calls, including breakdown of lower precedence calls when necessary to make trunks available for higher precedence calls. This capability was demonstrated in June on the two UTX-1200 switches at Lincoln Laboratory, including all the necessary features (dialed-in precedence level codes, preempt warning tones, and automatic call breakdown and trunk seizure). Another benchmark capability achieved in June was RCP-controlled site-to-site calling via the EISN satellite channel. This had required special effort because of CCS complications due to the broadcast nature of the satellite channel, and because of software modifications required in the Packet/Circuit Interface (PCI) equipment.

It was decided that a video tape would be prepared to document the performance of the RCP network, including prerecorded examples of various real-call experiments over satellite and terrestrial channels. A basic objective was that the tape be usable as a standalone presentation, including descriptions of the purposes and architecture of the network, so that (for example) it could be used in the future to quickly acquaint visitors with the EISN equipment in a laboratory setting. The tape was produced successfully, and its first major application was in the EISN Program Review at DCEC on 17 July. Copies of the tape were provided to the Army and RADC EISN participants and to DCEC at that time. The video tape was also an integral part of a pair of conference papers delivered in October 1985 (References 5 and 8), which undertook to give a concise summary of the EISN program and its accomplishments.

In late July, as described in Section 3, the two UTX-1200 switches at Lincoln were shipped to RADC along with their associated RCPs and interface cabinets. After installation and integration of this equipment, it was used to carry out a set of in-house real- and phantom-call experiments at RADC. The culmination of this work was multiswitch voice call setup with multilevel precedence and preemption, as well as phantom calling experiments with emulated traffic, at the 26 September briefing and demonstration at RADC described below.

2.2.2 Phantom-Call Experiments

In order to validate the RCP system implementation with larger network topologies and more realistic traffic loads than could be supported with the few available real trunks and switches, a phantom-call capability was added to the system design. Phantom calls do not make use of real trunks or switches in the RCP network. Instead, they use phantom trunks whose number can be specified more or less arbitrarily. However, they do require real CCS channels between RCP nodes for signaling purposes. They are offered to the network by simulator modules in each RCP and are handled by the same RCP call sequencer that handles real calls. Because there is no switch interaction, the handling of phantom calls is somewhat simpler, and setup times are shorter than for the case of real calls, but the CCS signaling protocol is identical.

A network of RCP processes interconnected with CCS links constitutes a distributed simulation environment in which experiments can be carried out to study the effects of different routing and/or preemption algorithms, adaptation to changing traffic loads, etc. Such a distributed environment complements the Call-by-Call Simulator by offering a more detailed, closer-to-the-real-world simulation. However, because of its real-time nature, the network of RCPs is limited in the size of network and density of traffic that it can simulate.

In this section we describe the capabilities of the RCP distributed network simulator and present the results of some experiments we have carried out to test those capabilities.

2.2.2.1 Simulation Capabilities

The simulator module in the RCP provides both for generating phantom-call traffic dynamically and for running a precalculated file containing a list of calls to be offered in sequence when the simulation is started. In both cases the simulator in a particular RCP is concerned only with phantom traffic offered from that site. It is therefore necessary to specify the traffic at each node in the simulated network independently.

Dynamic traffic generation is controlled via RCP menus that allow a user to specify a traffic matrix that contains, for each destination and precedence level, an average traffic rate in erlangs. An independent load factor that multiplies the matrix elements allows runs to be made at different overall traffic levels without changing the relative traffic balance. Menu items also specify average call duration and the total run time for the simulation as well as allowing a choice of statistical distribution for the call durations and offer rates. The choices are "deterministic," "uniform," and "exponential." Choosing "exponential" for both distributions gives traffic statistics similar to those used in the Call-by-Call Simulator. Since specification of all parameters for an experiment is a relatively cumbersome process, another option allows them to be read from a file that can be prepared in advance using a text editor.

Precalculated call files are text files specifying for each call a destination, a precedence, a duration, and an offer time in milliseconds from the start time of the simulation. Such files can be prepared manually using a text editor, and such a procedure is useful in creating small test files, but their greatest value is in allowing comparison with simulations carried out using the Call-by-Call Simulator. That program produces call files that can be reformatted to serve as RCP call files. We have used such reformatted files extensively in our experiments.

Simulations are started by menu commands, and options are provided to start immediately or at a specified time in the future. Since it is often a requirement that simulations start at more than one node at essentially the same time, the latter option is a useful one. However, for cases when multiple-node experiments are controlled via the multiplexing facilities described in Section 3, the immediate start option is preferred, since the multiplexer provides a better synchronous start than does the clock-time start, particularly if the experiment involves multiple machines whose clocks are likely to differ.

6

As a simulation proceeds each RCP accumulates statistics on the network as seen from its node and can produce a complete trace of the history of all calls it has processed. The statistics include tallies of all offered and carried calls by precedence level. For calls originating at each RCP, tallies are further broken down by destination node. Statistics are also accumulated on CCS message traffic.

The simulation results may be viewed on the RCP terminal as they accumulate, and/or written into files for off-line analysis. The writing of files may be done periodically, at the start and finish of an experimental run, or on command from the terminal.

The PDP-11/44 processors that support the RCP programs are large enough to run several RCP nodes in a single machine. The number of such virtual nodes that can be supported in a single processor is limited by the memory available in the processor; the number of ports available for CCS signaling use; the size of process tables, etc. in the operating system; and the speed of the CPU. Since the RCP software assumes that the processor is keeping up with real time, it is necessary to adjust the combination of network complexity and offered traffic load so that the processing does not fall behind real time on the average. There is also a need to keep the peak CCS signaling traffic low enough so that the buffering capacity in the operating systems is not overloaded. Excessive signaling rates will cause CCS packets to be lost, and even though the CCS protocol provides for retransmission in such situations, lost packets will produce artifacts in the simulation that may distort the results.

Experience has shown that a single PDP-11/44 can support four virtual RCPs with an overall traffic load of the order of 100 erlangs of 3-min. average duration calls without serious distortion of the call setup time statistics. To run experiments with heavier traffic loads, we scaled the call durations and the times between call offerings in proportion to the traffic increase so that the CPU load and CCS traffic remained approximately the same as they would have been at the 100-erlang load. On occasion, using scaling in the other direction, we ran experiments with lighter loads or fewer nodes at faster than real-time rates.

The setting up of a phantom-call experiment involves physically connecting the CCS ports (tty ports on the processors) and preparing "configuration files" for each virtual RCP node. The configuration file provides information about the topology of the experimental network in the form of routing tables and link parameters (e.g., identity of neighbor nodes and trunk capacities). The routing tables give an ordered list of alternate routes to all possible network destinations. Each route entry contains a parameter that indicates the minimum number of links that must still be traversed if the call is to reach the destination using that particular routing choice at that point in its path. This parameter is used in the routing algorithm to halt further exploration of alternate routes when the best remaining path is longer than the allowable limit. A menu choice allows the experimenter to set a limit on the number of links beyond the minimum that a call is permitted to use.

The configuration file also provides information used by the RCP to generate a graphical representation of the network for display on the terminal, and it specifies whether or not trace information will be written into files as the experiment progresses.

7

The RCP programs provide for a choice between two kinds of call preemption. The simpler of the two, called "Blind-Back," preempts the oldest of the lowest precedence calls occupying a trunk on a link needed to route a higher precedence call. The "Back" in "Blind-Back" indicates that the preemption occurs in the backward, or acknowledgment, phase of the CCS setup protocol. As a result, a call is preempted only for cases in which a successful path for the preempting call has been found. The other preemption option, called "Guided," also works on the backward phase. It chooses, among the lowest precedence calls, the oldest of those that have the most links in common with the preempting call. With both preemption options, the algorithm will favor preemption on the shortest path over exploring alternate routes.

2.2.2.2 The Simulation Environment

When simulation experiments were first started there were three PDP-11/44 RCP processors at Lincoln Laboratory. Accordingly, we specified an experimental network of nine virtual RCPs with three running in each of the three processors. The test network was defined based on a subset of the DSN1 network that had been explored using the Call-by-Call Simulator. Link capacities were chosen to fit the limit of a maximum of 115 trunks/node set by memory space in the RCP implementation. Call files were generated by the Call-by-Call Simulator using a scaled partitioning of the DSN1 traffic matrix. Figure 1 shows a graphical representation of the 9-node network. The numbers adjacent to the links show the number of phantom trunks assumed in the simulation.
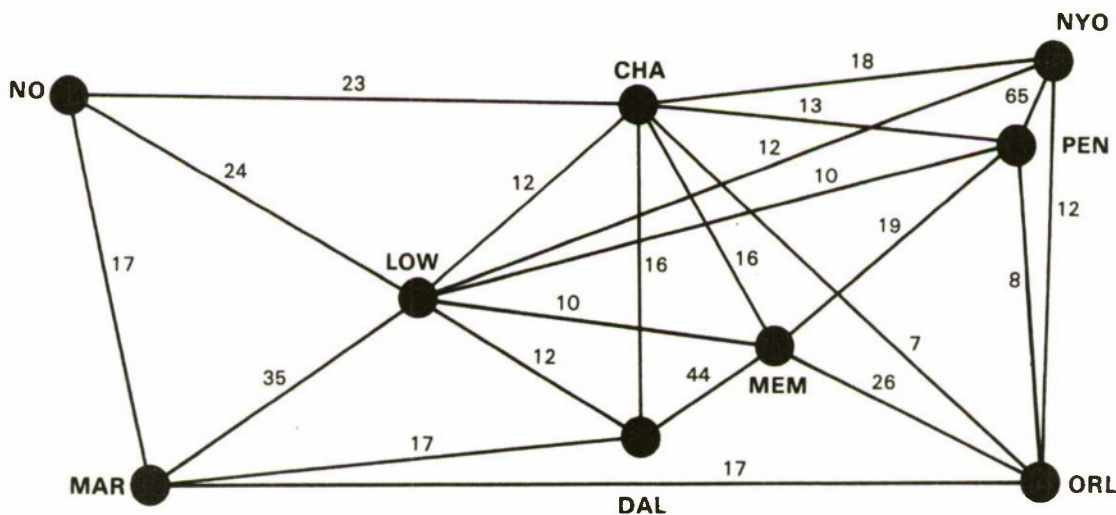
Figure 1.   The 9-node simulation network.

8

The physical arrangement for the experiment is shown in Figure 2. The experiment was controlled from a single terminal that could be switched to any or all of the virtual RCPs using the capabilities of the "todown" and "mpx" programs described in Section 3 of this report.

A number of experiments were run using the 9-node network, and results were compared with those from the Call-by-Call Simulator. While there was approximate agreement between the simulations, there were significant differences in detail. These were analyzed and found to be due primarily to two problems. The first was that the RCP call sequencer did not handle correctly the "glare" situation that develops when the RCPs at the two ends of a link both attempt to use the last trunk in the link at the same time (i.e., within a small time window). The second was a protocol error that caused calls to block spuriously because the RCPs at the ends of a link did not always see a trunk becoming free at the same time.

The above-mentioned problems were corrected in new versions of the RCP software, but two of the three processors were shipped to RADC before the new versions were operating satisfactorily. Consequently, further experimentation has had to make use of a smaller 4-node network (Figure 3) that could run in the one remaining processor. The node names and trunk capacities for this network were arbitrarily chosen. The results discussed in the following section were obtained with this smaller network.
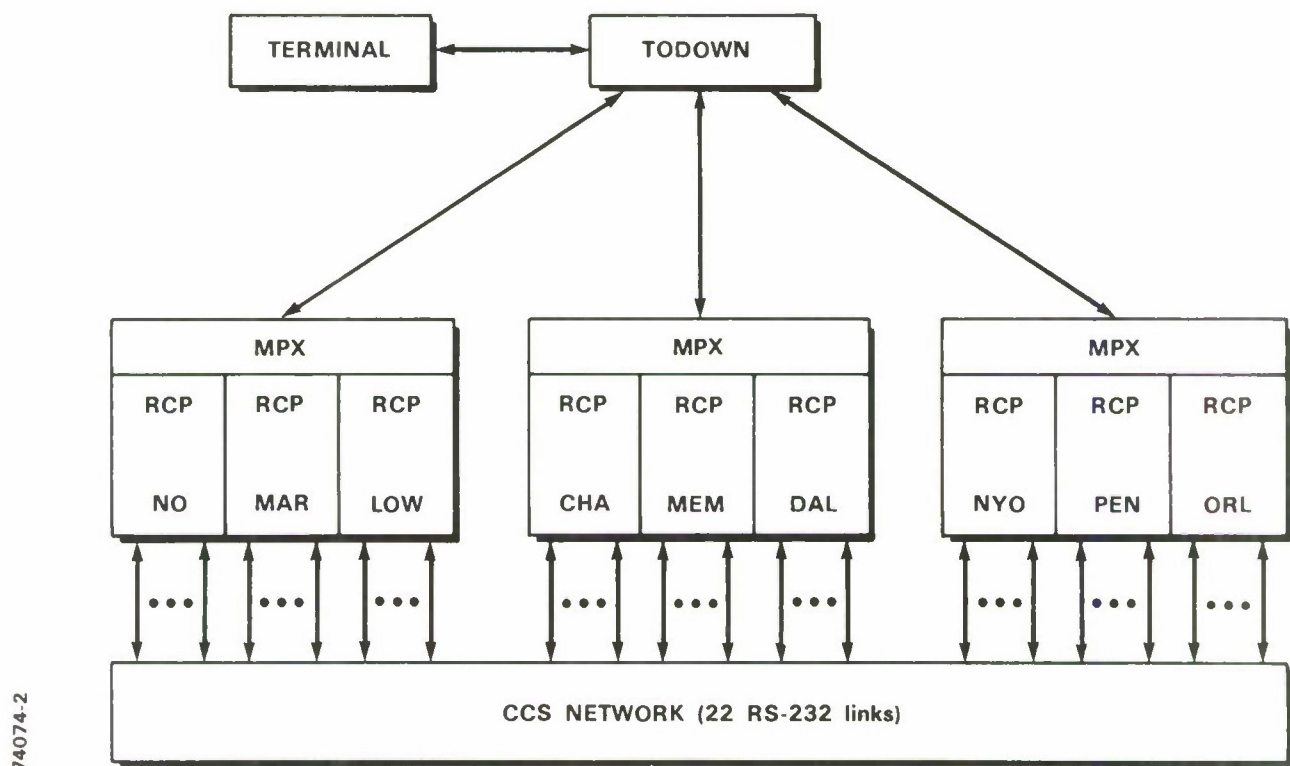
Figure 2. The 9-node simulation environment.

9
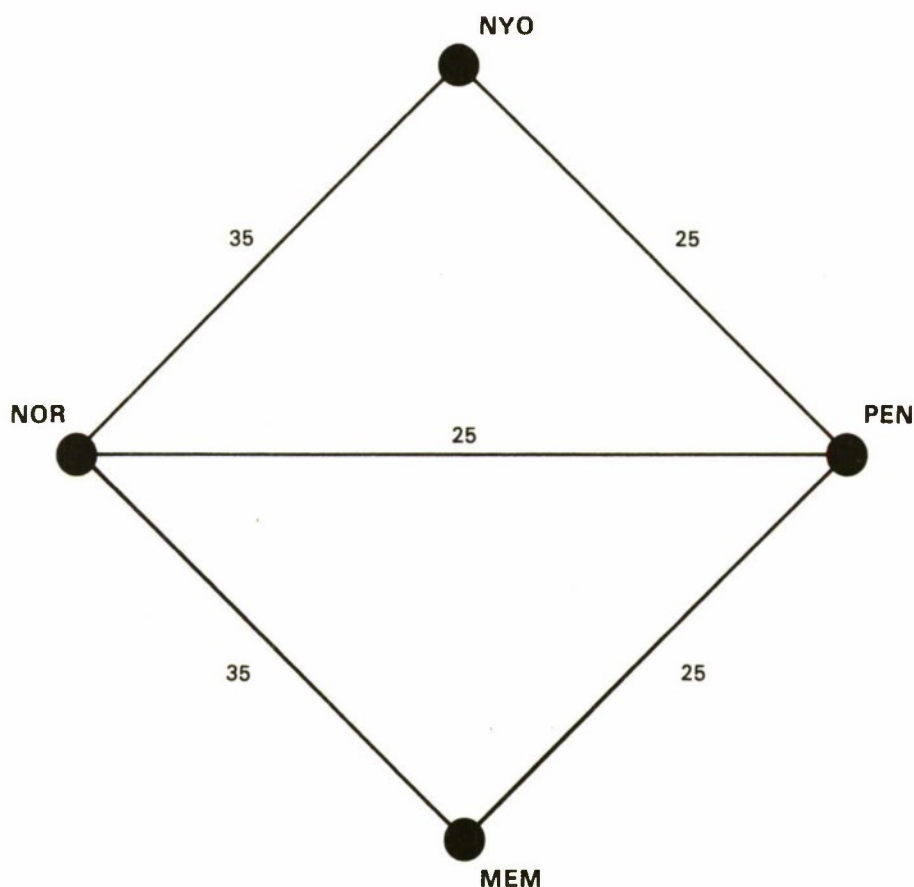
Figure 3.   The 4-node simulation network.

2.2.2.3 Simulation Results

Tests of the 4-node network were all run using the symmetric traffic matrix shown in Figure 4. The matrix represents an overall load of 100 erlangs which results in a relatively low level of blocking for the trunk capacities shown in Figure 3. The traffic was divided between two precedence levels with 60 percent at the lower level. To explore the behavior at higher traffic levels, the matrix was multiplied by factors of 2 and 3 with the running time scaled to keep the rate of offered calls constant, so that the results would not be distorted by overloading of the processor.

In the tests, two routing algorithms were compared. The first restricted calls to paths of minimum length as measured by the number of network links in the path. The second algorithm allowed calls to use extra links if the minimum path was blocked. Low-precedence calls were allowed one extra link. High-precedence calls were allowed two extra links, but the second link was never used due to the small size of the 4-node network.

10

	NOR	MEM	PEN	NYO
NOR	0	5	10	5
MEM	5	0	10	10
PEN	10	10	0	10
NYO	5	10	10	0

Figure 4.   Test traffic matrix for the 4-node network (in erlangs).

The results presented here are all from test runs that used the "Blind-Back" Preemption option. Preliminary runs with the "Guided" Preemption option did not show any significant improvement relative to the "Blind-Back" option. Such a result is to be anticipated because the path lengths in our 4-node net are not long enough for much benefit to be gained from picking the call with the most links in common with the preempting call.

Figure 5(a) shows the blocking observed in the test runs. As can be expected, the blocking is a little lower for the extra link algorithm for the cases of lower overall load. However, at high overall traffic the extra link algorithm exhibits higher blocking because the calls that utilize extra links tie up more network resources than those confined to minimum paths.

Figure 5(b) shows the fraction of the low-precedence calls that were preempted to carry the high-precedence calls. Here, the minimum path algorithm always gives better performance.

Since the results from the RCP network and the Call-by-Call Simulator are quite close, the differences are best seen in a numerical format. Figure 6 shows the comparison. The differences are due to a random component in the times that it takes the RCP network to process call set-ups and hang-ups. Because of this randomness there are a few call sequences that may be handled differently by the RCP net than by the simulator. When two calls contend for a single remaining trunk in a link, the outcome will depend on which one arrives first unless they differ in precedence. The simulator processes calls in the exact order that they are generated, but the RCP network may see the same list of calls in a slightly different order due to clock differences between computers and time-sharing delays in processors serving more than one virtual RCP node. For most of the calls in an experiment run the timing differences have no effect because there are no other calls contending for resources within the window caused by the random timing effects. Occasionally, however, the random nature of the call generation algorithm causes enough call requests to bunch together to result in contention being resolved differently between the simulator and the RCP network.

11
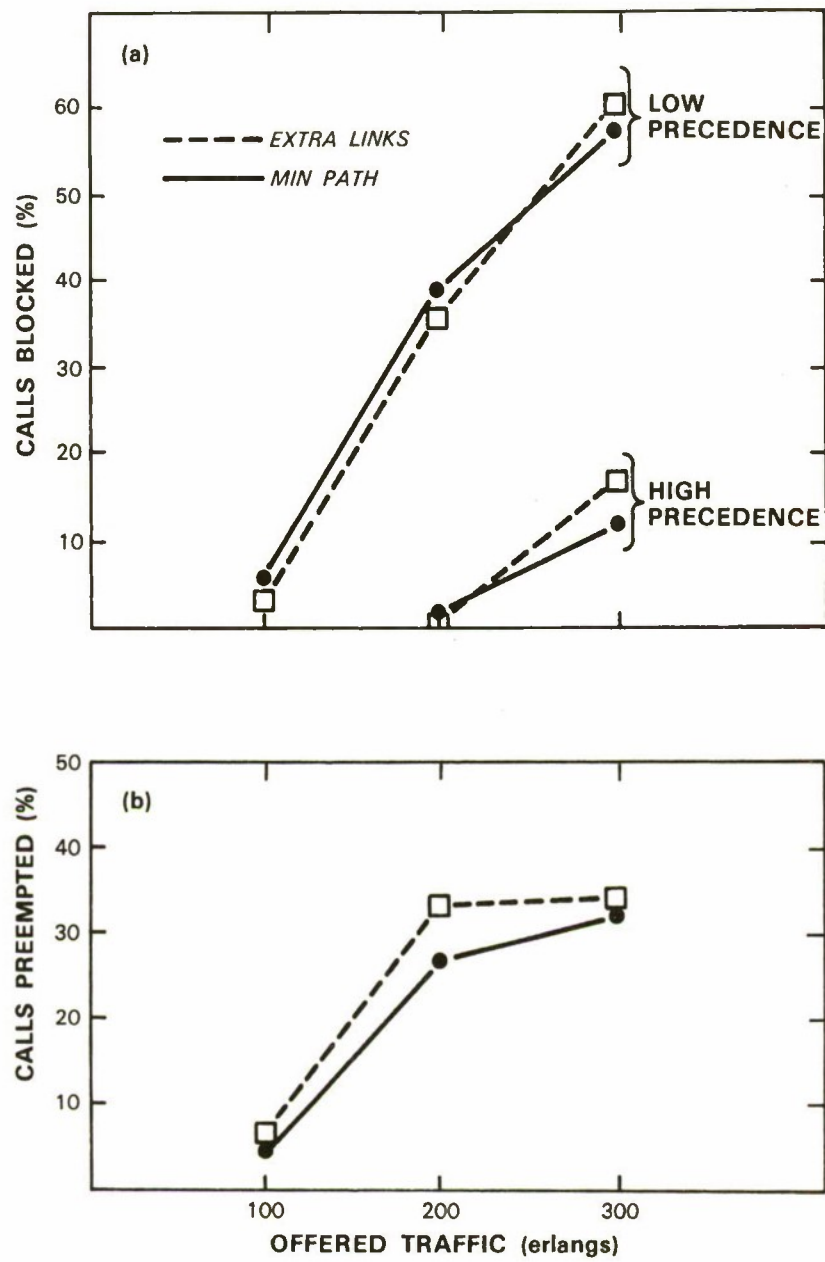
Figure 5. Test results for the 4-node network: (a) Fraction of calls blocked and (b) fraction of calls preempted.

12

PRECEDENCE	ROUTING	LOAD	% BLOCKED		% PREEMPTED	
			RCP NET	SIMULATOR	RCP NET	SIMULATOR
LOW	MINIMUM PATH	100	6.1	6.4	4.4	4.4
		200	38.5	38.3	27.1	27.6
		300	57.5	57.2	32.1	32.8
	EXTRA LINKS	100	2.8	2.9	6.6	6.5
		200	35.5	35.5	33.8	34.1
		300	60.2	59.5	34.5	34.8
HIGH	MINIMUM PATH	100	0.0	0.0		
		200	0.9	0.9		
		300	11.9	12.6		
	EXTRA LINKS	100	0.0	0.0		
		200	0.1	0.1		
		300	17.2	18.0		

Figure 6. Comparison between RCP 4-node network emulation and Call-by-Call Simulator results for the same network.

The kind of timing effect that can make a difference can be seen in the following scenario which we observed in early tests using the 9-node network. Consider the situation depicted in Figure 7. Here, the numbers next to the links represent the number of free trunks remaining in the links between the nodes. We have two calls to consider. Call A from node NO to node NYO cannot succeed because there is no free trunk between CHA and NYO. However, this fact is not known at node NO, so that node will commit the last trunk on the link to CHA to Call A and send a "Setup" CCS packet to CHA for Call A. When CHA gets the packet it will be unable to handle the call to NYO because there is no trunk available, and will send back a "Setup Fail" packet to NO. In the meantime, if Call B is offered at NO, it too will fail because the trunk to CHA has been committed to Call A. If Call B is offered just a little later, it will succeed because the failure of Call A will have become known at NO, and the trunk to CHA will be available for Call B. We thus have a situation in which a small difference in the time required to handle one call can affect the fate of another, and of course the effect can extend further to calls that do not themselves arrive in a critical time window. If Call B succeeds, it may occupy its trunks for many minutes thereby causing other calls to block, which would not do so if Call B had failed initially.

Because of the timing sensitivity just discussed, repeated RCP network emulations did not yield exactly the same results. There tended to be differences in detail from run to run, but we did not observe any timing effects that made significant differences in overall blocking or preemption averages. Runs using the RCP traffic generators, with nominally the same average offered traffic but with different random call sequences, showed greater run-to-run variations than did repeated runs with the same call sequence. However, it should be noted that if care is not taken to avoid CPU overload during periods of peak activity in RCP network emulations, time window effects can become significant by causing call events to bunch together unrealistically.

13
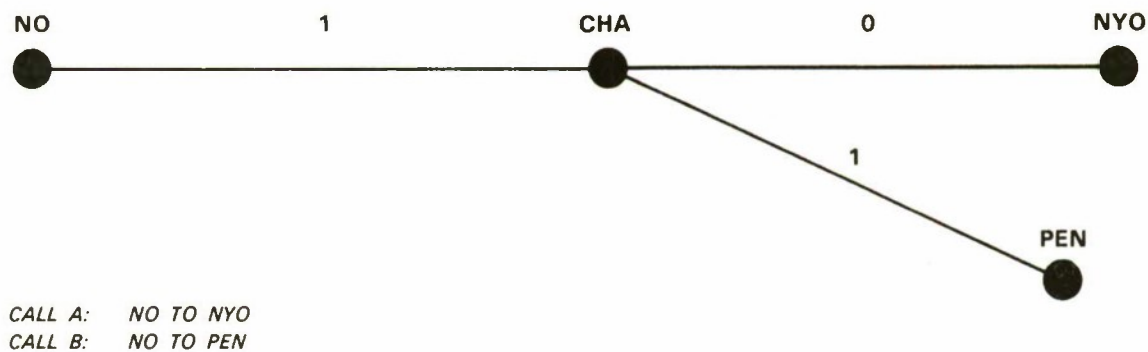
Figure 7. Example call scenario in which the success of Call B depends on its time relation to Call A.

The RCP network emulation tests also provided measurements of Common Channel Signaling (CCS) traffic for the protocol in use. Figure 8 shows those measurements as well as a comparison with calculated values from the Call-by-Call Simulator. There are two factors that contribute to the differences between the RCP results and those from the simulator. The RCPs use RS-232 links for signaling, and consequently, the signaling packets must all be multiples of 8-bit bytes in length. The simulator calculates the minimum number of bits needed to handle the protocol information. In addition, the RCPs exchange "fill-in" messages to keep the links active when there is no real traffic. These messages add to the measured traffic for the RCPs but are not included in the simulator calculations.

2.2.2.4 Conclusions and Discussion

The phantom-call experiments have demonstrated the ability of the RCP implementation to simulate networks with realistic traffic loads. The test bed at RADC will have three RCP processors; and with that facility, networks with interesting topologies can be explored with phantom-call experiments.

The CCS traffic results show that the RCP routing algorithms and related protocols make only modest demands on CCS channel capacity. These results confirm earlier estimates obtained using the Call-by-Call Simulator.

The close agreement between the results from the Call-by-Call Simulator and those from the RCP network emulation serves to validate both implementations. The RCP test bed is closer to a real network than is the simulator. It actually goes through all the steps of setting up and taking down calls following the CCS protocol steps in detail. The simulator operates at a higher level of abstraction, and by avoiding many details can speed up the simulation so that large, complex networks can be studied efficiently. There is always some concern, however, that some detail ignored or incorrectly dealt with by a simulator may have an important effect on the results obtained.

The value of having a test bed such as the RCP network, as well as a simulator, lies in the fact that the test bed exposes the effects of implementation details on system performance. For
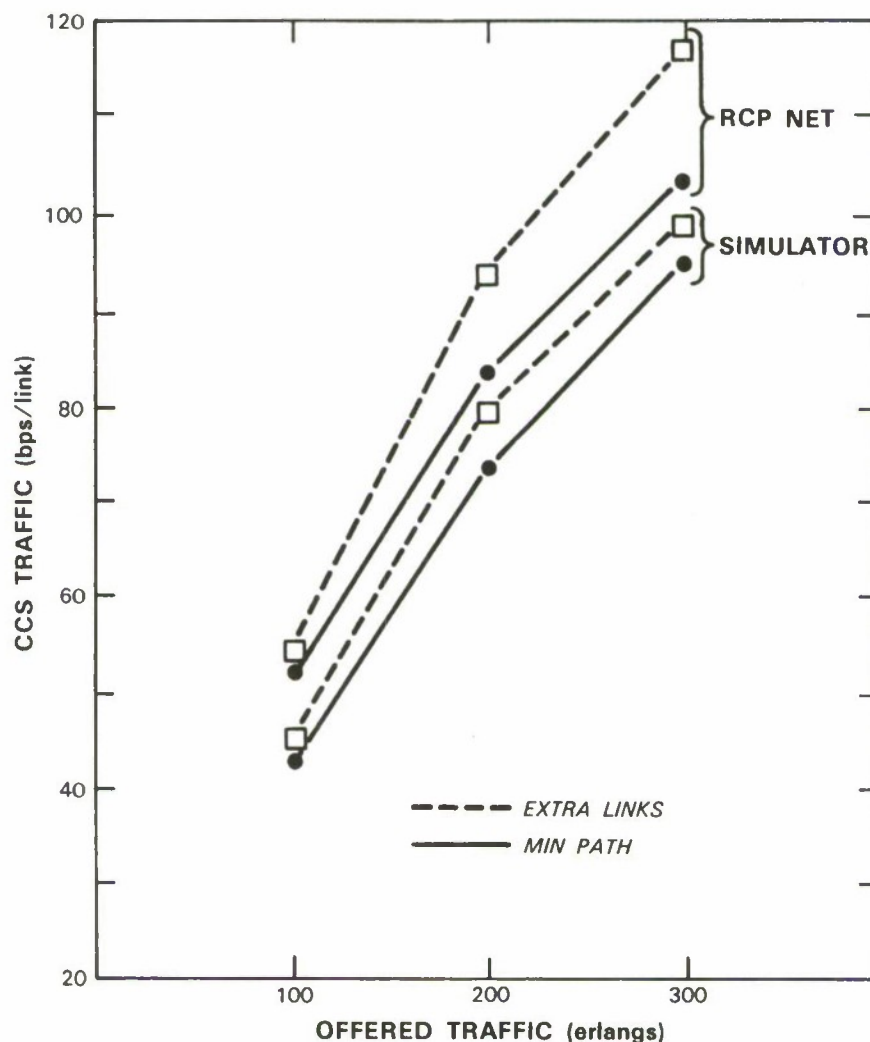
Figure 8. Test results for the 4-node network: Common Channel Signaling (CCS) traffic.

example, the call scenario discussed above in relation to Figure 7 represents a situation in which incorrect handling of a detail affected system performance. When we first began comparing the detailed behavior of the simulator with the RCP network, we observed that the simulator was not allowing for the time that Call A in that scenario would tie up the trunk while waiting for the CCS message to indicate failure of the call. In that version of the simulator, Call B would always succeed no matter how soon after Call A it was offered. As a consequence, runs with the simulator would show a few more calls succeeding than would runs with the RCP network. The effect on overall blocking was not great because in any one run there were only a few calls that were offered within the time window and under situations where there was contention for the last trunk. Still, the simulator results were in error, and as a result of this observation, a change was made in the simulator to correct its handling of this case. It is unlikely that the error would have been discovered without the existence of the test-bed implementation.

15

2.3 CALL-BY-CALL SIMULATOR DEVELOPMENT

A Call-by-Call Simulator has been developed to permit study of the dynamics of complex new routing and preemption procedures. The simulator was begun in FY82, and has been used extensively. Over the past three years, new capabilities have been added to the simulator and more complete documentation has been provided. The simulator has been in regular use at DCEC as well as at Lincoln; the latest version was delivered to DCEC in September 1985.

The simulator now includes all the new routing procedures developed during the EISN program, including (a) Spill-Forward Mixed-Media Routing; (b) Mixed-Media Routing with Single-Stage Crankback; (c) Mixed-Media Routing with Remote Earth-Station Querying; (d) Mixed-Media Routing with Full- and Limited-Search Originating Office Control; (e) Adaptive Mixed-Media Routing; (f) Precedence Flooding (flood all high-precedence calls); and (g) Precedence-Blocked Flooding (flood only blocked high-precedence calls). The simulator also includes four reference routing algorithms: (a) POLYGRID Routing as used in CONUS AUTO-VON; (b) Forward Routing (only route forward to a node that is closer to the destination); (c) Modified Forward Routing (same as Forward Routing, but route forward to a node that is directly connected to the destination); and (d) Primary Path Only Routing (no alternate routing). A selection of three preemption procedures is also available: (a) Blind-Out Preemption as used in CONUS AUTOVON; (b) Blind-Back Preemption (only preempt after a complete call path has been found); and (c) Guided Preemption (preempt the fewest calls on the shortest path to the destination). All preemption procedures can be used with a ruthless trunk search algorithm (preempt on any link listed in the routing table if no free trunks are available) or with a friendly trunk search algorithm (preempt only after examining all links listed in the routing table). Other options in the simulator allow selection of network management controls (precedence-dependent call path length limits, routing table length limits, and satellite hop limits); call retry behavior; type and amount of network damage; plot output; timing of runs to produce detailed statistical printouts; and level of detail to provide in-trace printouts.

A block diagram of the simulator is presented in Figure 9. Input files on the left contain controls for the current run; routing tables produced by an external program; a description of the network topology and link capacities; and the offered traffic matrix. A call file produced by an external call generation program can be substituted for the file containing the offered traffic matrix. This substitution is more efficient when making multiple simulation runs with identical offered-call patterns, because calculations of Poisson sequences of calls are performed only once. Output files on the right in Figure 9 contain statistical tables and data that can be used to create plots of results. The logic contained in the Routing and MLPP Processor of the simulator implements all the new routing and preemption procedures. Other simulator software modules collect statistics, produce output files, simulate transmission of Common Channel Signaling (CCS) information between switches, and initiate internal events (damage, new calls, call retries, and call terminations). Simulator documentation includes a user's guide, internal documentation files, and demonstration runs.

16
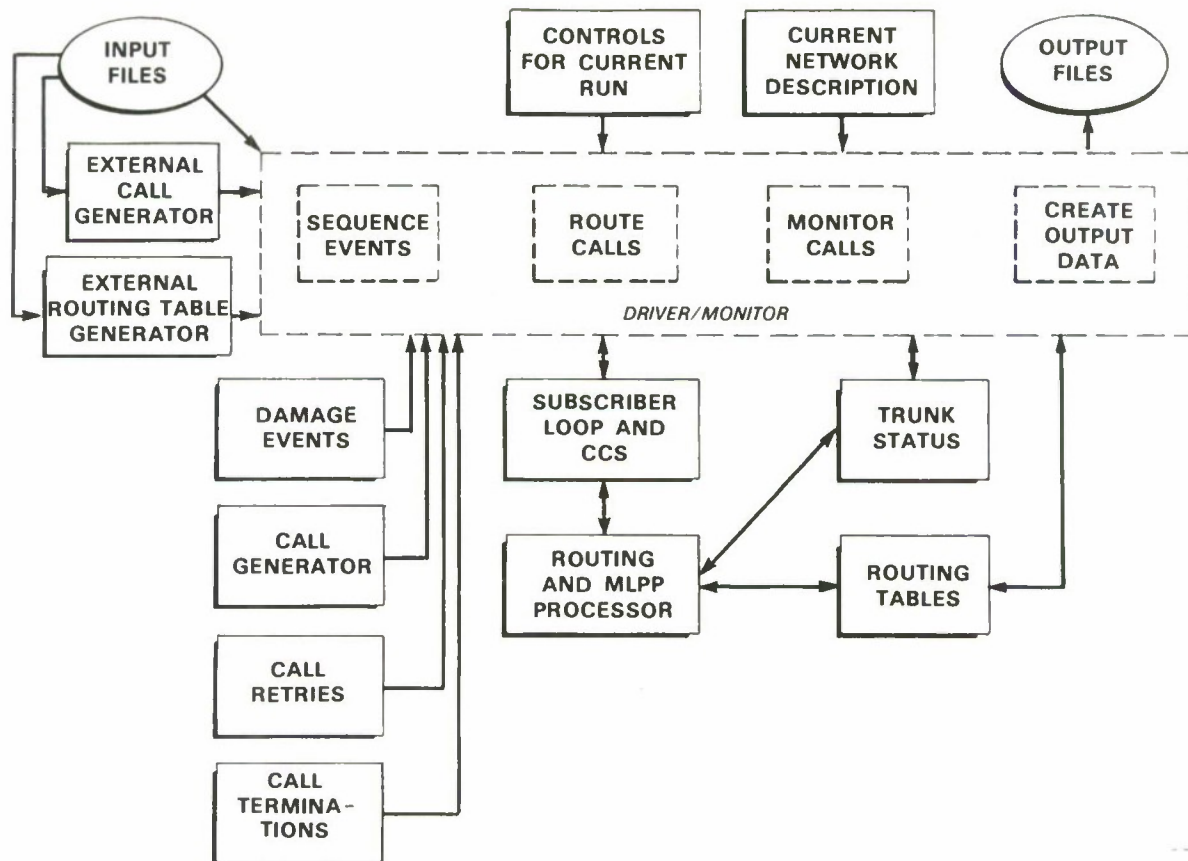
Figure 9. Block diagram of Call-by-Call Simulator.

The simulator consists of roughly 27,000 lines of RATFOR code and 200 subroutines. RAT-FOR was selected because it is a modern structured language that is automatically translated into portable FORTRAN code. The portability of the simulator has been demonstrated during the past year when runs were performed at Lincoln, at DCEC in Washington, at GTE in Phoenix, and at GTE in Massachusetts. At Lincoln the simulator has been run on a VAX computer under the UNIX operating system using FORTRAN 77, and on an IBM 3081 and an Amdahl 470 computer under the IBM CMS operating system using FORTRAN IV. At other sites it has been run using various IBM computers under the TSO and CMS operating systems, using FORTRAN IV and FORTRAN VI.

Major new capabilities added this past year include POLYGRID Routing, Originating Office Control Routing, efficient data base access routines, point-to-point satellite links, and improved crankback algorithms which function when the projected call path length is too long.

17

POLYGRID routing was requested by DCEC, and is an important baseline routing procedure to use for comparison purposes because it is currently used in CONUS AUTOVON. It was not possible to add POLYGRID Routing until recently, when a program was developed at DCEC to generate POLYGRID Routing tables. The simulator implements the version of POLYGRID routing that is described in report DCAC 370-V120-1, "CONUS AUTOVON Routing Philosophy," dated 27 February 1981. Different trunk search strategies are used for low- and high-precedence calls, inside and outside the home grid of the destination node. Rotating triples are used in an attempt to search different routes during call retries, by varying the order of routing table entries between successive call attempts.

Originating Office Control Routing was also requested by DCEC as a reference routing procedure, because it is used in the European Telephone Network and in the Pacific. Two types of Originating Office Control Routing are included in the simulator. With Limited-Search Originating Office Control Routing, calls are allowed to examine alternate routes only from the source and from special spill nodes. When a call is blocked, it is cranked back to the originating node or to the most recent spill node encountered. Another path with a different first link may then be tried from the originating or spill node. With Full-Search Originating Office Control Routing, calls are allowed to examine alternate routes at all tandem switches. When a call is blocked, however, it is cranked back to the originating node or to the most recent spill node encountered.

Point-to-point satellite links were also requested by DCEC to model networks which use such links for long-distance connectivity. Point-to-point satellite links are treated exactly the same as land links in the simulator, except that the allowed number of satellite hops is limited.

An improvement was made in the method used in the simulator to determine whether call paths were too long, thereby making better use of the path-length information contained in routing tables. Call path length is normally limited to the sum of the number of links in the shortest path from the source to the destination, plus a number that varies with the call precedence level. Previously, a call was blocked if this limit was reached and the next node was not the destination. This technique, however, does not exploit the fact that each switch knows the length of the shortest path from itself to the destination. The old technique was improved by comparing the path length limit to the sum of current length plus the minimum projected length to the destination from the current node. In addition, this test was moved to the section of code that determines whether links in the routing table can be used. These modifications reduce the CCS communications load by reducing the number of calls that travel down dead ends and are blocked. They also improve performance with the friendly trunk-search algorithm, because higher-up routing table entries with free trunks will not be selected if they lead to excessively long paths. Instead, lower routing table entries will be used with preemption. Performance using Crankback and Originating Office Control was also improved during these modifications, by initiating Crankback not only when a call is blocked because no trunks are available, but also when a call is blocked because the path is too long. Prior to this, Crankback was not initiated when a call path was too long.

The simulator's running speed was increased by more than a factor of 4, by writing more efficient routines to maintain the call-in-progress data base and to select calls for preemption.

Run-time profiles performed after a number of simulation runs indicated that the previous versions of these data-base handlers had accounted for more than 80 percent of the CPU cycles used. These handlers had been written for simplicity but had never been optimized. First to be rewritten were the handlers used to insert and extract calls in the calls-in-progress data base and to find the next call to terminate. The new handlers maintain a doubly linked list of calls, sorted by termination time, with a free list and pointers into the active list that split the list into balanced bins. This structure reduces the average number of elements searched to insert a new call from n/2 to the square root of n, where n is the number of calls in progress. For a typical run with 3600 calls in progress, the average number of items searched is thus reduced from 1800 to 60. Handlers used to find preemptable calls were also rewritten. Other new routines maintain a linked list for each link in the network for use in preemption, containing the call numbers of all calls using each link. Only the list for a given link must be searched, instead of the complete calls-in-progress data base, whenever a call on a link must be preempted. With these changes, the five most heavily used subroutines in the simulator now take roughly 4 to 6 percent of the CPU time. Subroutine usage is now well-balanced, and further increases in speed will be difficult to achieve. The VAX CPU time currently required to run an hour-long simulation with 20,000 calls, using Spill-Forward Mixed-Media Routing, is roughly 20 min..

New utilities have been added to the simulator this year to simplify the task of setting up simulation experiments. In previous years, utility programs were created to generate routing tables and call files. The utilities developed this year were to (a) compute offered traffic statistics from call files; (b) print the non-default control parameters selected for a run; (c) print the default values of control parameters; (d) print link statistics as defined in the link definition file; (e) create a new connectivity file that can be used to create a routing table after damage for adaptive routing; (f) examine entries in the routing table; (g) compute offered traffic statistics from the offered traffic matrix; and (h) convert POLYGRID Routing tables to a format that can be used with the simulator.

Numerous small changes were also made to the simulator in preparation for the final delivery. The switch-processing delay added at each switch to compute the call setup time was made a run-time parameter. All included files were named using a uniform convention. Definitions of all symbolic constants were put into one file, to reduce the work of changing array sizes for runs with large networks. The detail trace printout was modified to make it easier to follow. Subroutine comments and headers were updated. The simulator user's guide was updated. Finally, node names were added to node numbers in the output statistical tables.

2.4 ROUTING STUDIES

An extensive series of simulation runs was performed to compare all routing procedures under normal conditions, and with different types of damage and different patterns of offered traffic. More than 600 runs of the simulator, taking from ten CPU minutes to more than three CPU hours each, were performed on a VAX-11/780 to compare more than twenty types of routing procedures. During each run, data were collected over 1 h of simulated time using the final

version of the Call-by-Call Simulator. Only average data from the final 45 min. of each run were used for comparisons.

2.4.1 Test Networks

Comparisons were performed using the three networks shown in Figures 10, 11, and 12. Characteristics of these three networks are presented in Table I. Network POLY20 in Figure 10 is a 20-node network with low connectivity (average links per node) and with no satellites. Network DSN1 in Figure 12 is a 20-node network with 1 satellite, 5 Earth stations, and higher connectivity than POLY20. Network CONUS is a 54-node network with no satellites that is representative of the current CONUS AUTOVON network.

Network POLY20 was designed using a POLYGRID network design program provided by DCEC. Node locations and traffic patterns were selected from a 45-node network provided by DCEC, having characteristics similar to those of the current CONUS AUTOVON network. POLY20 consists of the 20 nodes having the highest offered traffic, along with their offered traffic values. These data were input to the POLYGRID design program set to produce a network with as few links as possible, with average blocking on links of 0.36. A POLYGRID Routing table for this network was created using the POLYGRID design program. This routing table was converted to Lincoln format using a utility program provided with the final version of the simulator.
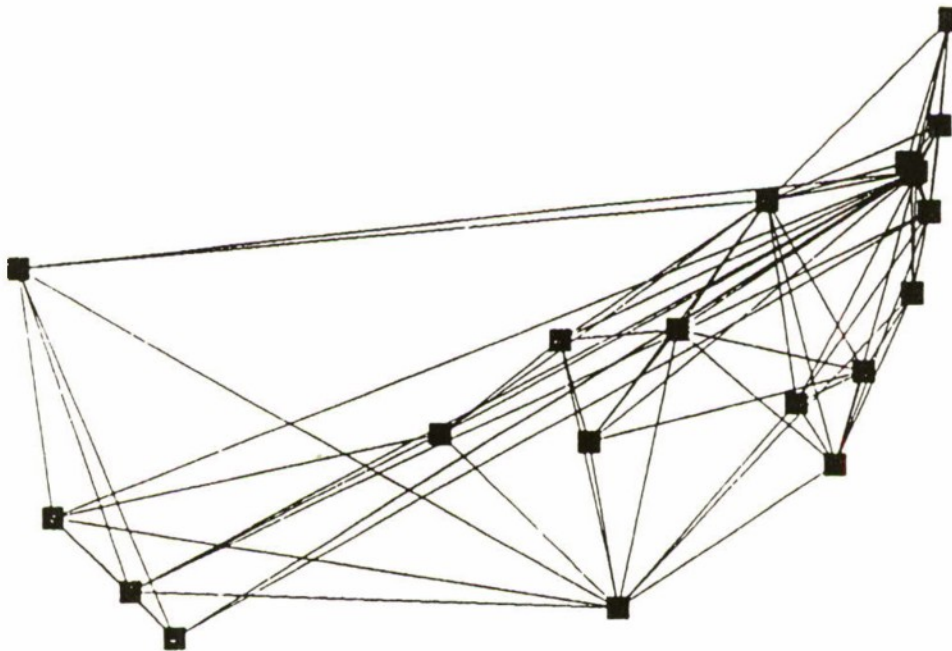
Figure 10. Test network POLY20.

20
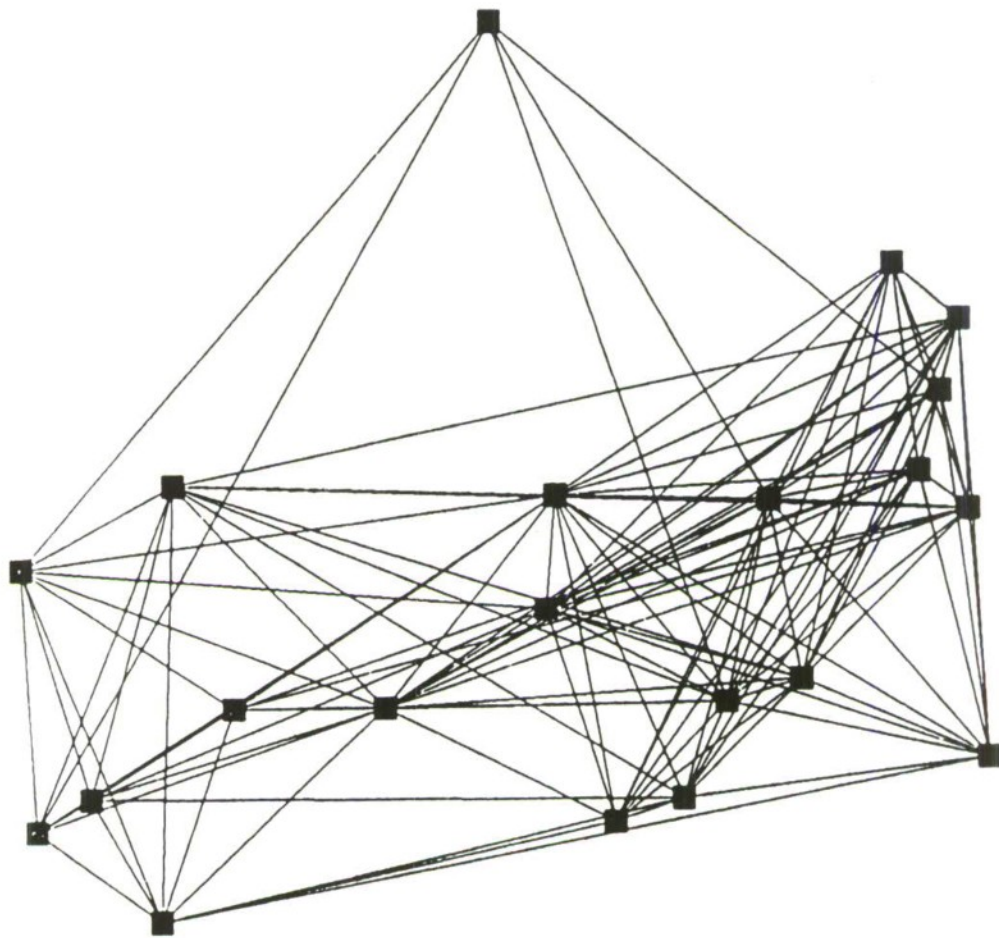
Figure 11. Test network DSN1.

21
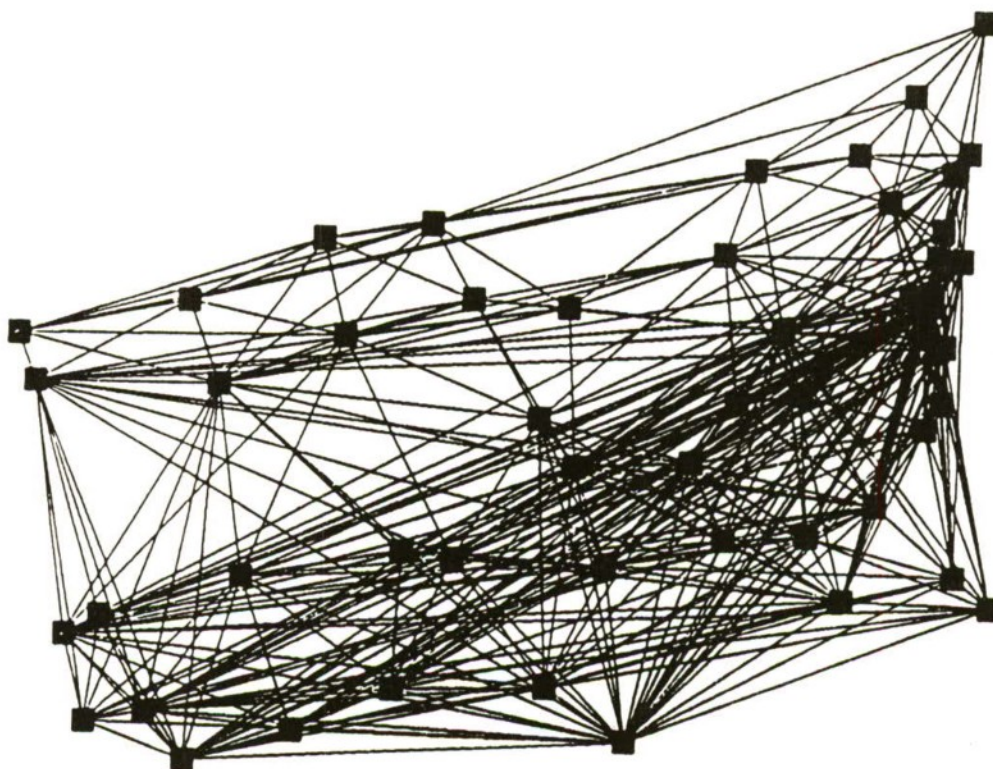
Figure 12.   Test network CONUS.

TABLE I
Characteristics of Three Test Networks

Network Name	Nodes	Sat	Earth Stations	Land Links	Avg Conn	Land Trucks	Total Traffic (erlangs)
POLY20	20	0	0	75	7.5	5359	3476
DSN1	20	1	5	118	11.7	1992	1518
CONUS	54	0	0	446	16.5	9153	6467

22

Network DSN1 has been used extensively in the past and is the only network in these comparisons which contains both land and satellite links. It is a minimum-cost network designed for a link-blocking probability of 0.1 and designed to route roughly one-third of all traffic over the satellites under normal conditions. It includes 20 switches, one DAMA satellite, and five Earth stations, and has a relatively high average connectivity.

Network CONUS is a 54-node network with node locations, trunk group sizes, and offered traffic similar to those in the current CONUS AUTOVON network. The POLYGRID Routing table used was similar to one created by AT&T for CONUS AUTOVON.

2.4.2 Routing and Preemption Procedures

Twenty-one different combinations of routing and preemption procedures were tested. These were selected to examine new and interesting features and also to provide reference comparisons to existing procedures. A list of the combinations used is presented in Table II.

As can be seen in Table II, the techniques of Mixed-Media Routing, Adaptive Mixed-Media Routing, and Flooding were compared to POLYGRID Routing, Modified Forward and Primary-Path Only Routing. All routing techniques were not used with all conditions and all networks. Precedence flooding was not examined with the large CONUS network because of the heavy CPU usage this required. Some of the other routing procedures that examine different preemption techniques, trunk reservation, and the effect of varying call-path length were also not examined with the CONUS network. Adaptive Routing was only used under damage conditions where it is different from normal routing procedures. POLYGRID Routing was used only with the POLY20 and CONUS networks which were designed to be used with POLYGRID Routing. Remote Earth-Station Querying was only used in DSN1, which is the only network that had satellites.

Mixed-Media Routing procedures were compared to POLYGRID and Modified Forward Routing in land networks (POLY20 and CONUS), even though these networks do not contain a land/satellite mix-of-transmission media. These comparisons were performed because these new mixed-media procedures differ significantly from the older types of routing, even in networks without satellites. In such networks, Mixed-Media Routing procedures differ from POLYGRID and Modified Forward Routing in the method used to limit call-path length and prevent loops. Mixed-Media Routing procedures limit call-path lengths explicitly, while POLYGRID Routing and Modified Forward Routing allow the call-path length to grow indefinitely, as long as the call continues to progress toward the destination. Mixed-Media Routing should thus tend to utilize network resources more efficiently under overload and with chaotic traffic patterns, when call path lengths with POLYGRID and Modified Forward Routing might be excessively long. In addition, Mixed-Media Routing should provide better control of routing flexibility because limits for call paths can be explicitly adjusted for each precedence level. Another difference between Mixed-Media Routing and POLYGRID Routing is in the method used to generate routing tables. Mixed-Media Routing tables are easy to generate and update because they do not have to

23

TABLE II

Routing and Preemption Procedures

Condition	Description
acb	Same as cb but adapt routing tables after damage.
acpbf	Cb with pbf and adapt routing tables after damage.
amm	Same as mm but adapt routing tables after damage.
aopbf	Ooc with pbf and adapt routing tables after damage.
aooc	Same as ooc but adapt routing tables after damage.
apbf	Same as pbf but adapt routing tables after damage.
apf	Same as pf but adapt routing tables after damage.
cb	Crankback calls. Same as mm but all calls crankback to previous node when blocked.
mfr	Modified forward routing.
mm	Spill-forward mixed-media routing with guided preemption. High-precedence call-path length limited to 2 more links than in the shortest path to each destination. Low-precedence call-path length limited to 1 more link. Ruthless preemptive trunk search. No call retries.
mmbb	Same as mm but blind-back preemption.
mmbp	Same as mm but blind preemption.
mmfp	Same as mm but friendly preemptive trunk search.
mmre	Same as mm but blocked calls retry.
mmrq	Same as mm but use remote Earth-station querying.
mmrt	Same as mm but reserve last two trunks in each link for high-precedence calls.
mmxl	Same as mm but maximum call-path length for high-precedence calls is 4 more links than in each shortest path.
ooc	Originating office control routing. Same as mm but all calls crankback to the source when blocked. A full search of routing table entries is allowed at tandem nodes.
pbf	Precedence blocked flooding. Same as mm but blocked high-precedence calls are routed using flooding.
pf	Precedence flooding. Same as mm but all high-precedence calls are routed using flooding.
poly	POLYGRID routing as used in CONUS AUTOVON.
polyre	Same as poly but blocked calls retry.
pp	Primary path only routing. No alternate routing.

explicitly disallow all paths that lead to loops. Loops are prevented during call routing by maintaining a trace of switches already visited and not routing back to a switch already in the call path. Alternatively, POLYGRID Routing tables are difficult to generate and update because all possible paths must be examined and paths that lead to loops and shuttles must be explicitly disallowed using route control digits.

Adaptive Routing was examined with many types of Mixed-Media Routing. The effect of adapting routing tables after damage was examined for Mixed-Media Routing (amm), and for Mixed-Media Routing with: Crankback (acb), Precedence Flooding (apf), Precedence-Blocked Flooding (apbf), Originating Office Control (aooc), the combination of Crankback and Precedence-Blocked Flooding (acpbf), and the combination of Originating Office Control and Precedence-Blocked Flooding.

A number of features available with Mixed-Media Routing were explored. Two types of Crankback were evaluated. Cb from Table II examines Single-Stage Crankback where all calls crank back to the previous node if blocked at a node. Ooc examines Originating Office Control Crankback, where calls crank back to the source when blocked. It uses a full search at each node, where all routing table entries are searched for an available outgoing trunk.

Two flooding techniques were compared. Pf from Table III examines Precedence Flooding, which routes all high-precedence calls using flooding. Pbf examines a more limited flooding technique which routes only blocked high-precedence calls using flooding.

Five types of preemption were compared. Mm examines Guided Preemption, which preempts the fewest calls on the shortest path to the destination after a call path has been found. Mmbb examines Blind-Back Preemption, which blindly selects any lower precedence call for preemption on each link where preemption is necessary, after a call path has been found. Mmbp examines Blind Preemption, which preempts blindly on each link as the call path is being set up. Mmfp examines a friendly, two-pass trunk search algorithm with Guided Preemption. This friendly search first searches all routing table entries sequentially for free trunks on a first pass, and before a second pass, which searches for trunks to preempt. A single-pass ruthless search, used with all other combinations, searches all routing table entries in sequence looking for either free or preemptable trunks during the first search. Pf and pbf examine a type of preemption only available with flooding. The number of links requiring preemption is used in the metric that compares call paths. This will result in the selection of call paths with the fewest links requiring preemption, whenever two or more paths have the same number of links.

The effect of call retries was examined to evaluate the "rotating triple feature" provided in POLYGRID Routing which alters the routing table order slightly between successive call retries. Call retries were examined using Mixed-Media Routing (mmre) and POLYGRID Routing (polyre). Under both conditions, 10 percent of all blocked calls were retried. The time to retry had an exponential distribution with a mean of 6 min.

Other features available with Mixed-Media Routing were also examined. The effect of Remote Earth-Station Querying was examined in network DSN1 under damage conditions

TABLE III	
Damage and Traffic Conditions	
Condition	**Description**
norm	Normal offered traffic, no damage.
dam1011	Damage 10% of land trunks, destroying largest links first.
dam4011	Damage 40% of land trunks, destroying largest links first.
dam10n	Damage 10% of nodes, destroying those with most offered traffic first.
dam40n	Damage 40% of nodes, destroying those with most offered traffic first.
dams	Damage satellite.
dams1011	Same as dam1011 but also damage satellite.
dams4011	Same as dam4011 but also damage satellite.
dams10n	Same as dam10n but also damage satellite.
dams40n	Same as dam40n but also damage satellite.
10over	10% traffic overload. Same traffic pattern as norm but all offered traffic values increased by 10%.
40over	40% traffic overload. Same traffic pattern as norm but all offered traffic values increased by 40%.
bhour1	Busy hour on the East Coast. Same as norm, but traffic within time zones starting on the East Coast and progressing to the West Coast multiplied by 1.0, 0.95, 0.9, and 0.8. Traffic between time zones multiplied by mean of values for different time zones. After multiplication, all traffic values normalized to maintain overall offered traffic at the value used for norm.
chaotic1	Chaotic traffic pattern. Each p-p traffic value randomly chosen to range between zero and twice the normal value, then all values normalized to maintain overall offered traffic at the value used for norm.
uniform	Uniform traffic pattern. Each p-p traffic value set to the same constant, chosen to maintain overall offered traffic at the value used for norm.

(mmrq). The effect of reserving the last two trunks in links for high-precedence calls was examined under all conditions (mmrt). Finally, the effect of allowing high-precedence calls to travel four more links than the shortest path between call source and destination, instead of two more, was examined under all conditions (mmxl).

Three routing procedures were included for reference purposes. Primary Path Only Routing (pp) was included to examine performance without alternate routing. POLYGRID Routing (poly) was included because it is currently used in CONUS AUTOVON. Modified Forward Routing (mfr) was included because it is similar to POLYGRID Routing, which is one of the simplest forms of alternate routing, and because it has been used extensively at DCEC for network analyses.

2.4.3 Analysis Conditions

Runs were performed under normal conditions, with different types of damage, with traffic overload, and with different patterns of offered calls. The effect of damage to nodes, links, and to the satellite was explored, and the effect of five types of deviant traffic patterns was measured. These patterns were designed to determine the performance of routing procedures under traffic overload; under mild perturbations of the offered traffic matrix, with expected busy-hour time-zone variations; and with major changes in the offered traffic matrix. Damage and traffic conditions are described in Table III.

All conditions in Table III were used with all test networks except those where the satellite was destroyed. The latter conditions could only be tested with network DSN1. Under each condition, the same sequence of offered calls was presented for each routing and preemption combination. Call sequences were calculated once, prior to runs, using the desired traffic pattern. Call durations and call interarrival times were exponentially distributed. Average call durations were set to 6 min. for the POLY20 and CONUS networks, and to 3 min. for the DSN1 network. The percentage of offered traffic with each precedence level was 80 percent routine and 20 percent priority in POLY20 and DSN1. In CONUS, percentages were 80 percent routine, 15 percent priority, 3 percent immediate, 1.5 percent flash, and 0.5 percent flash override. The total number of offered calls during a 1-h simulation run ranged from roughly 22,000 to 48,000 calls under normal conditions.

2.4.4 Results

It would be difficult to present results for all combinations of routing and preemption procedures over all network conditions. Instead, the next two sections present results for all routing and preemption procedures for one network and one condition, followed by summary results for five routing procedures across all damage and traffic conditions.

2.4.4.1 Performance with Network Damage

Representative results for one network and one condition are presented in Figure 13. Graphs in this figure present results for POLY20 when 40 percent of the trunks are destroyed. The top
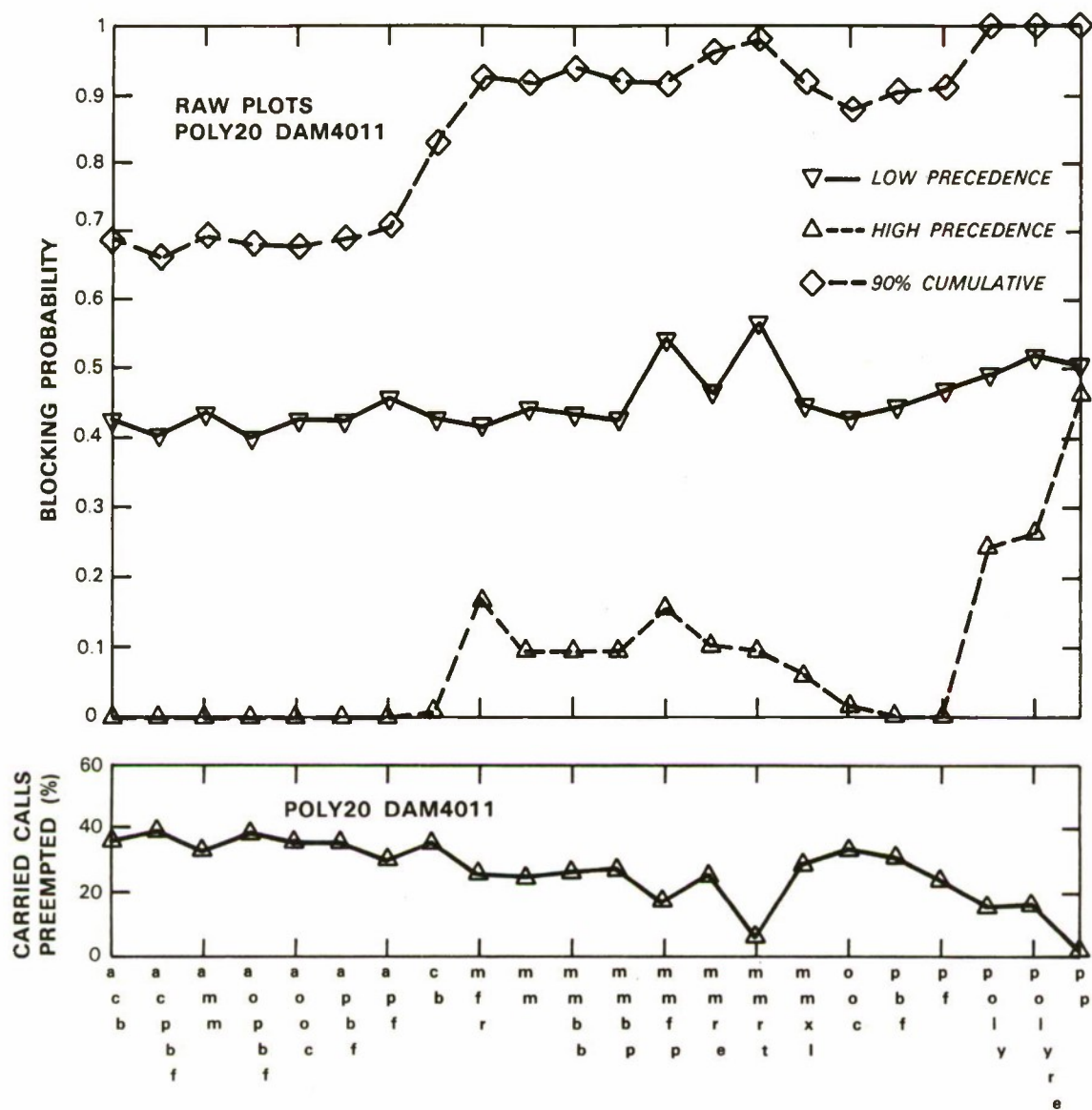
27

Figure 13. Results for network POLY20 when 40 percent of all trunks are destroyed.

74074-13

28

graph in this figure presents blocking probability results, and the bottom graph presents the percentage of carried calls that were preempted. Results are presented for twenty-two different combinations of routing and preemption procedures, running from Adaptive Routing with Crankback (acb) to Primary Path Only Routing (pp). The three curves in the upper graph present the blocking probability for high-precedence calls (short dashed line), the blocking probability for low-precedence calls (solid line), and the 90-percent cumulative blocking probability for all calls. The 90-percent probability provides an effective upper limit on the blocking probabilities experienced in the network. It was obtained from a histogram created by placing the amount of traffic offered between each node pair in a bin corresponding to the point-to-point blocking probability between the nodes. The cumulative distribution resulting from this histogram was used to determine the blocking probability exceeded by only 10 percent of the traffic in the network.

As can be seen in Figure 13, routing procedures differ substantially in high-precedence blocking, percent carried calls preempted, and 90-percent cumulative blocking levels. Poorest performance is provided by Primary Path Only Routing (pp) and POLYGRID Routing (poly). Blocking for both low- and high-precedence calls is high for both of these routing procedures. Best performance is provided by Precedence Flooding (pf). Precedence Flooding Routing preempts the fewest low-precedence calls among those routing procedures with almost zero blocking for high-precedence calls and roughly equivalent blocking for low-precedence calls. Almost equivalent performance is provided by Precedence Blocked Flooding (pbf) and almost all types of Adaptive Routing. All Adaptive Routing procedures provide lower than 90-percent cumulative blocking levels than nonadaptive procedures. Good performance is also provided by Mixed-Media Routing procedures using Crankback (cb), and Originating Office Control (ooc).

Other minor differences between routing procedures can also be noted. For example, Friendly Preemption (mmfp) degrades performance for both high- and low-precedence calls. Trunk Reservation (mmrt) degrades performance for low-precedence calls, does not change performance for high-precedence calls, and reduces the number of calls preempted. Guided Preemption used with Mixed-Media Routing (mm) preempts slightly fewer calls than Blind-Back Preemption (mmbb) and Blind Preemption (mmbp). However, the differences are small. Allowing high-precedence calls to have longer path lengths (mmxl) improves performance slightly for high-precedence calls with little degradation for low-precedence calls. Although the effect of call retries is small, POLYGRID Routing (polyre) handles retries no more effectively than Mixed-Media Routing (mmre). Similar results were obtained in other networks when land links were damaged.

Summary results for a more restricted set of five routing procedures in all networks and across all damage conditions are presented in Figures 14 through 16. The upper graph in each figure presents the blocking probability for high- and low-precedence calls for each type of routing. The lower graph presents the percentage of carried calls that were preempted with each routing procedure. As can be seen, performance degrades as 10 percent and then 40 percent of the land trunks are destroyed in these networks. Performance does not necessarily degrade, but varies, when 10 percent and 40 percent of the nodes are destroyed because the traffic from these nodes is eliminated from the network. Differences between the performance of different types of routing were greatest when 40 percent of the land trunks were destroyed.
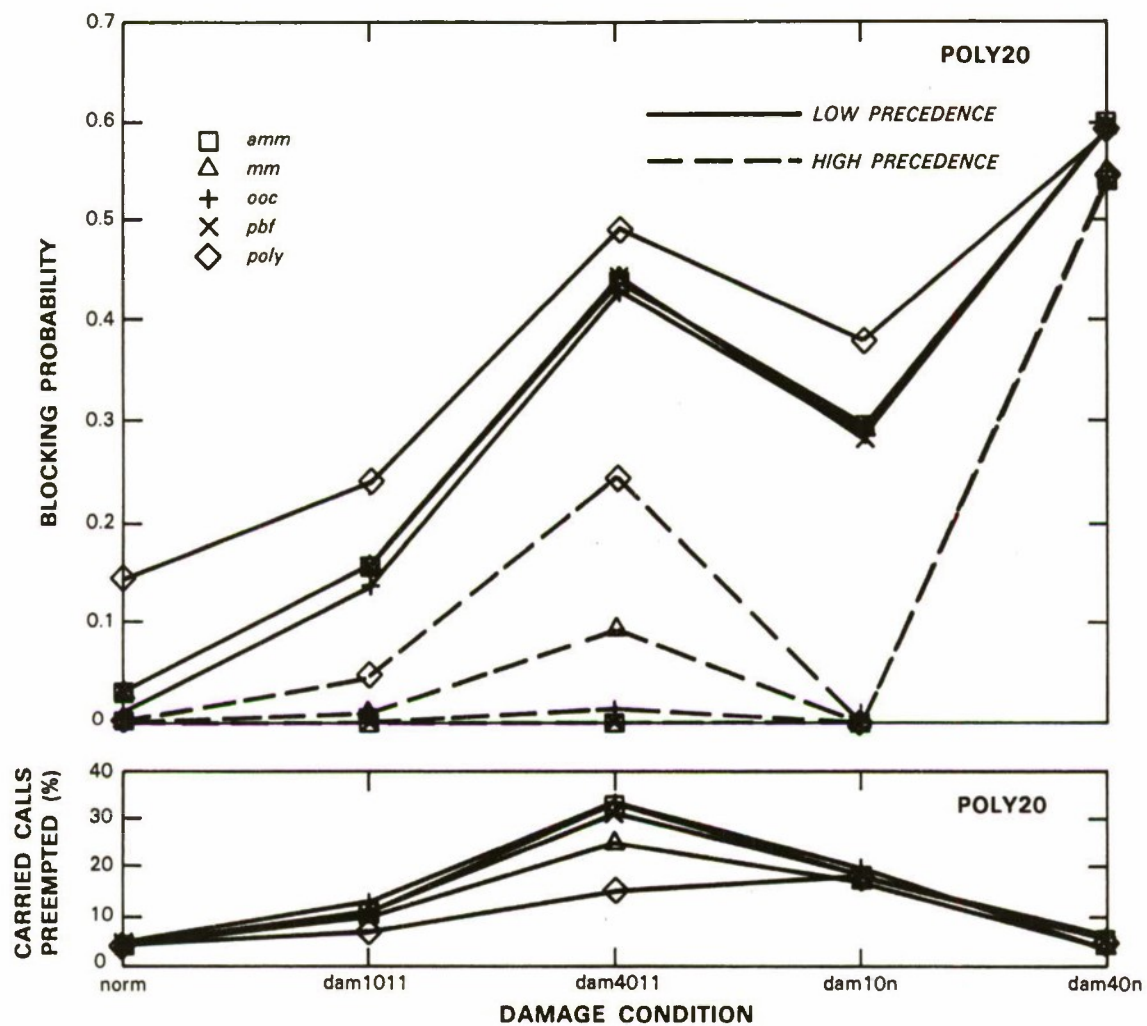
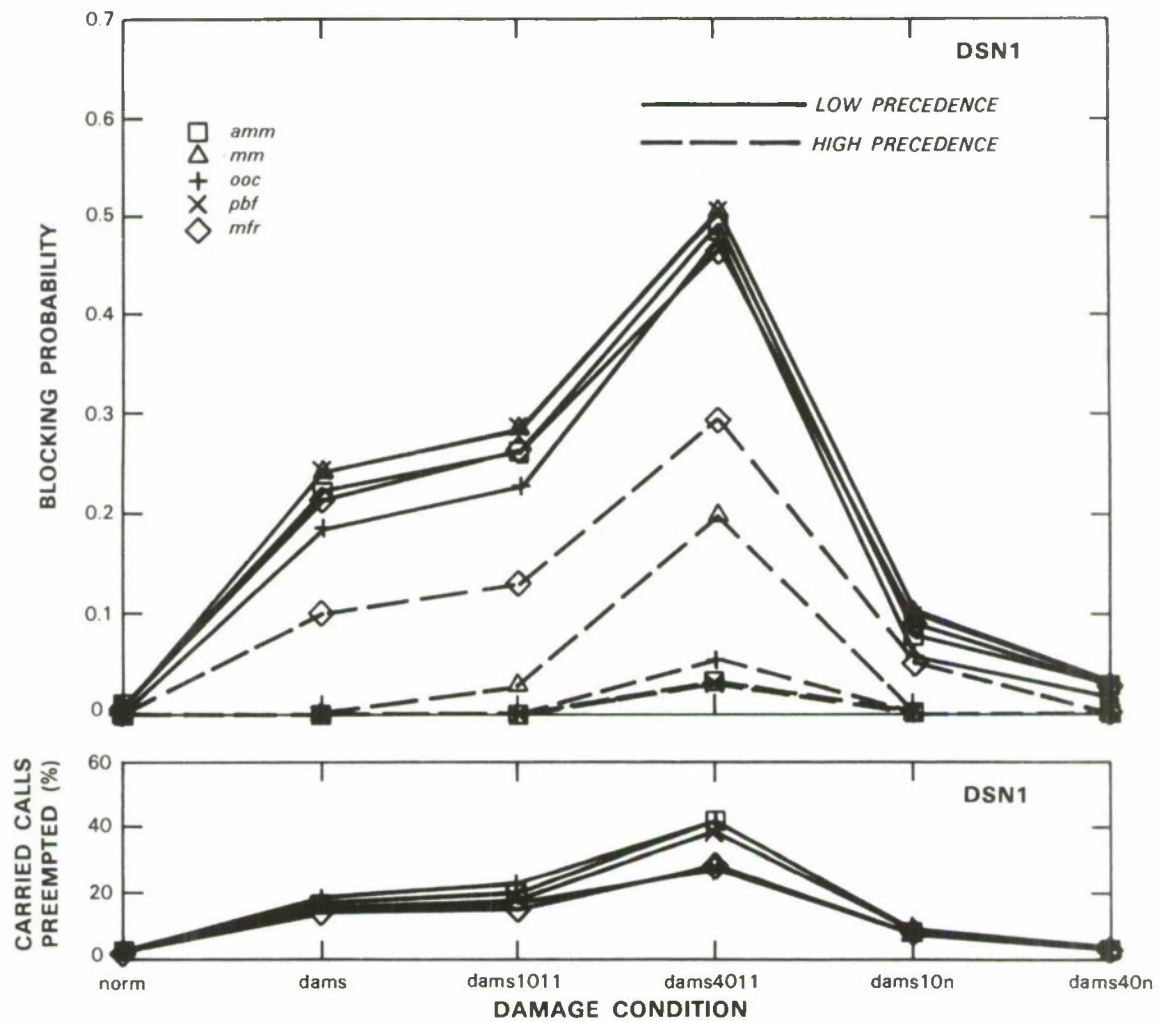Figure 14. Results with damage in network POLY20.

74074-14

30

Figure 15. Results with damage to the satellite, land links, and nodes in network DSN1.
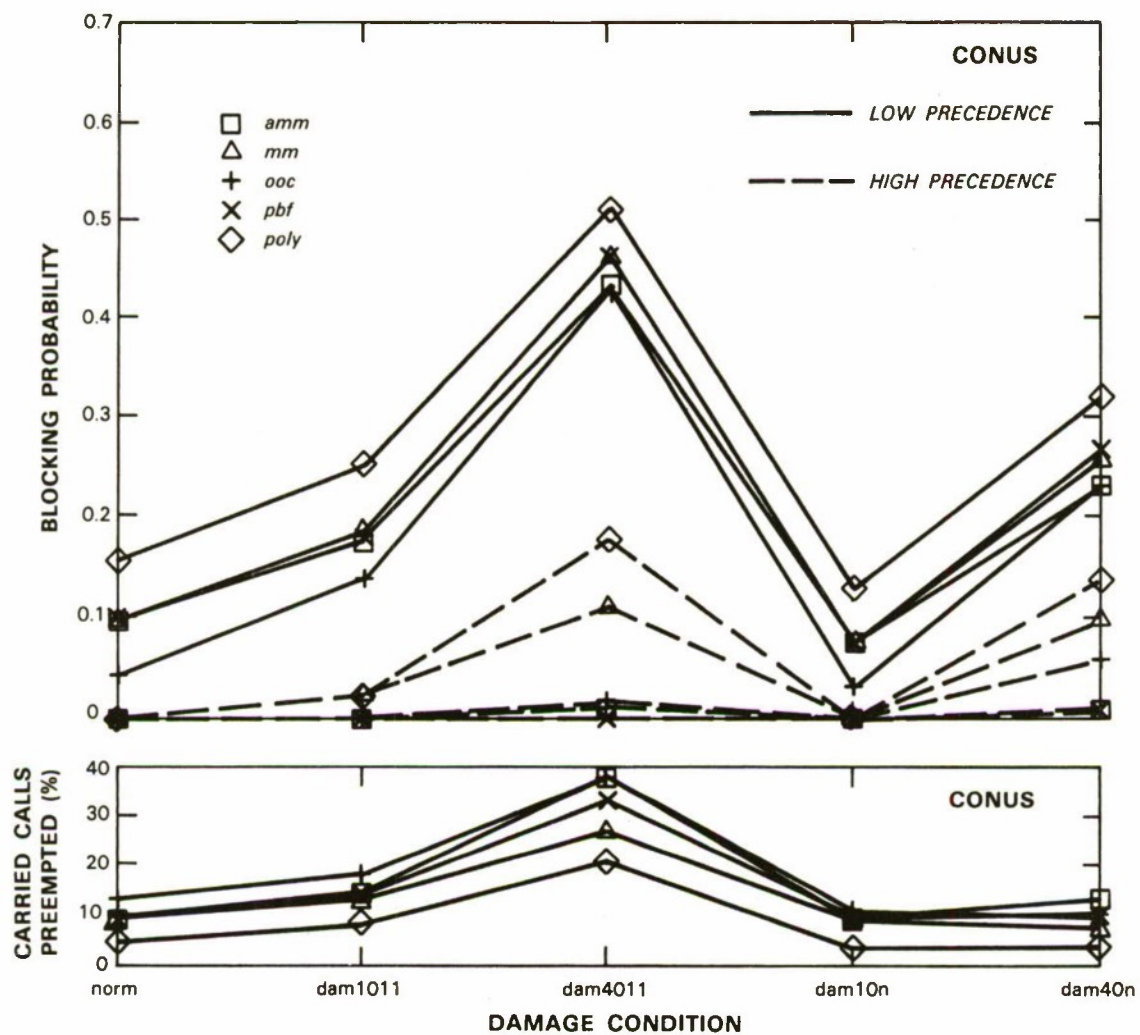
Figure 16.   Results with damage in network CONUS.

Poorest performance was provided by POLYGRID Routing and Modified Forward Routing. These techniques resulted in the highest blocking for high-precedence calls under almost all damage conditions. Best performance was obtained using Precedence-Blocked Flooding, Adaptive Mixed-Media Routing, and Mixed-Media Routing with Originating Office Control. Mixed-Media Routing was intermediate. Among the top three types of routing, Precedence-Blocked Flooding tended to result in fewer preemptions, and Originating Office Control tended to provide lower blocking for low-precedence calls.

2.4.4.2 Performance with Different Traffic Patterns

Representative results obtained with different traffic patterns are presented in Figure 17. This figure contains summary results for network CONUS for a set of five routing procedures across all traffic patterns. The top graph presents the average number of calls in progress in CONUS for high- and low-precedence calls. This is a more meaningful metric than blocking probability under network overload when the number of offered calls varies. The lower graph again presents the percentage of carried calls that were preempted.

Results obtained under overload demonstrate that the new routing procedures which provide better performance with network damage do not degrade performance under overload. The total number of calls in progress increased under overload with all new routing procedures. This is an important result because the increased routing flexibility provided by complex routing procedures could lead to excessively long path lengths under overload which reduce network utilization and degrade performance. It is evident from Figure 17 that this did not happen. The average number of calls in progress for high-precedence calls increased in proportion to the number of calls offered for all routing procedures because all high-precedence calls could be routed with preemption. Primary-Path Routing, Mixed-Media Routing, and Precedence Blocked Flooding carried the most low-precedence calls; the total calls carried increased for all new routing procedures under overload.

All new routing procedures also performed well under varying traffic conditions (bhour1, chaotic1, uniform) while Primary-Path Only Routing provided poorest performance. Among those routing procedures that performed well, Mixed-Media Routing with Originating Office Control carried slightly more low-precedence calls.

2.4.5 Summary

New routing and preemption procedures were compared under conditions of network damage, overload, and different traffic patterns. New routing procedures performed better than POLYGRID Routing and Modified Forward Routing under almost all conditions. The improvement in performance was greater under network damage and with aberrant traffic patterns. New procedures successfully routed more high-precedence calls and provided a more compact distribution of point-to-point blocking across node pairs for all calls. The improvement in performance for high precedence calls was generally accompanied by increased preemptions.

33

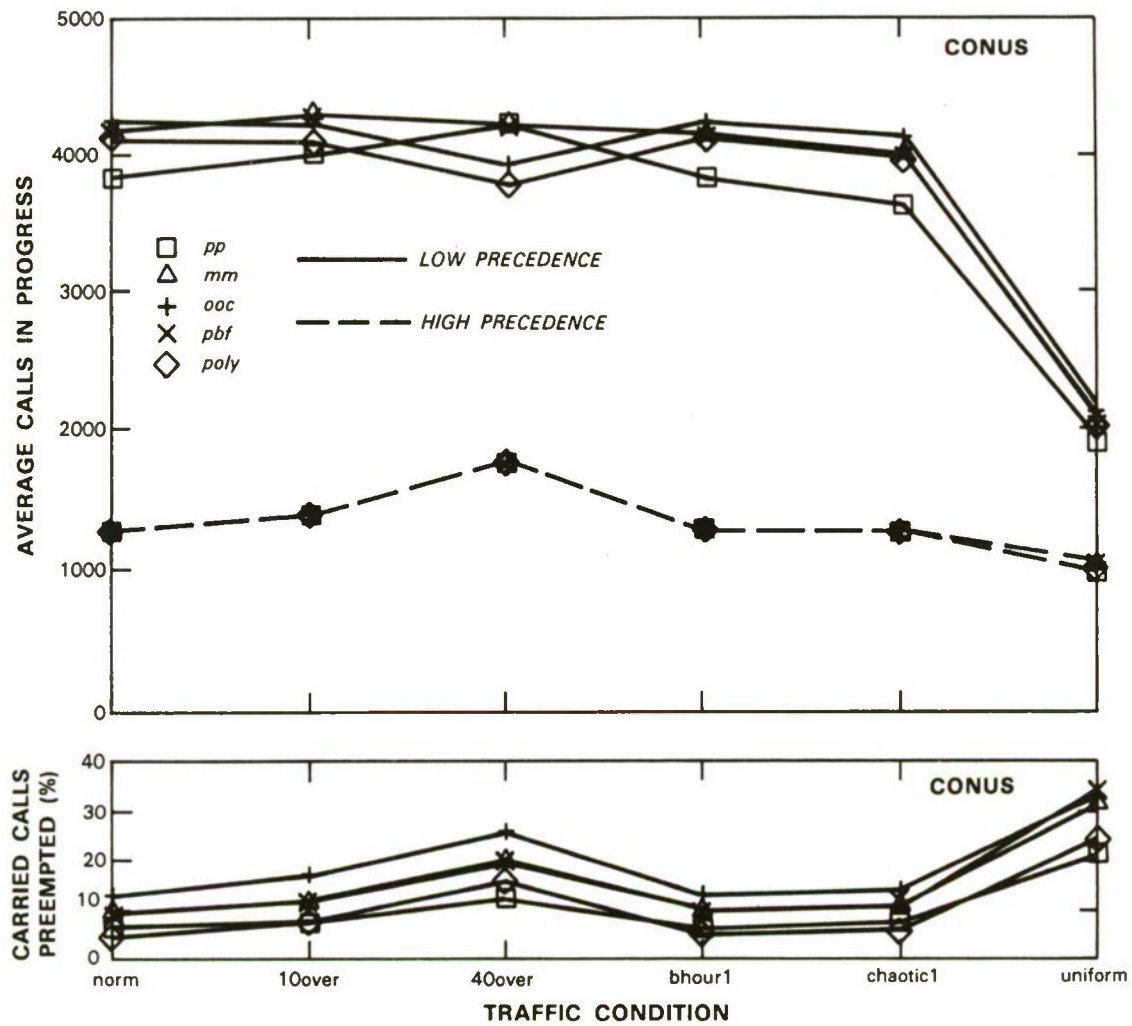*Figure 17. Results with different traffic conditions in network CONUS.*

74074-17

34

Greatest improvements with new routing procedures were obtained under damage conditions when land links were destroyed. For example, in 20- and 54-node POLYGRID networks, with 40 percent of all land trunks destroyed, the average number of calls in progress for high-precedence calls was 20 to 32 percent greater with new routing procedures (Adaptive Mixed-Media Routing and Mixed-Media Routing with Crankback and with Guided Preemption) than with POLYGRID Routing. In network DSN1 with both land and satellite links, 16-percent more high-precedence calls were carried with new routing procedures than with Modified Forward Routing. The difference increased to 37 percent when, in addition, the satellite was destroyed. The percentage traffic provided acceptable service and also improved with new routing procedures. In the two POLYGRID networks when 40 percent of all trunks were destroyed, the percentage traffic provided unacceptable service (blocking above 0.8) dropped from roughly 24 percent with POLYGRID Routing to 3 percent with Adaptive Mixed-Media Routing. The 90-percent cumulative blocking probability dropped from 1.0 with POLYGRID Routing to roughly 0.67 with Adaptive Mixed-Media Routing. These improvements were accompanied by increases in the number of low-precedence calls preempted. In general, the number of additional low-precedence calls preempted was roughly equal to the number of additional high-precedence calls carried.

Among the new routing procedures, best performance was normally provided by precedence flooding. Flooding all high-precedence calls routed the most high-precedence calls successfully with the fewest preemptions. In addition, the average CCS transmission rate with precedence flooding was not excessive. In both 20-node networks, the average transmission rate with flooding never exceeded 500 bps. Precedence-blocked flooding, which used flooding only for blocked high-precedence calls, typically performed only slightly worse than precedence flooding. It, however, typically reduced the CCS bandwidth required by more than a factor of 2. In the 54-node POLYGRID network studied, the average CCS transmission rate for precedence-blocked flooding never exceeded 500 bps.

Good performance without flooding and without adaptation was provided by Mixed-Media Routing with Crankback and Originating Office Control. Both types of routing routed more high-precedence calls than Mixed-Media Routing alone, but did not degrade performance significantly under network overloads. Results with Crankback differ from previous preliminary results because calls now crankback whenever the projected call-path length is too long, because call path lengths are carefully limited, and because both high- and low-precedence calls were allowed to crankback.

Adapting routing tables after damage typically only provided a small reduction in the number of high-precedence calls that were blocked when used with Mixed-Media Routing with Crankback or Originating Office Control. It did, however, decrease the traffic that provided unacceptable service. The upper curve in the top graph of Figure 13 illustrates this improved performance. In this curve, the 90-percent cumulative blocking level is the lowest for those routing procedures which use adaptation. Improvements with adaptation are again normally accompanied by an increase in the number of calls preempted.

New preemption procedures were found to perform slightly better than existing procedures. The reduction in number of carried calls preempted was typically 5 to 10 percent when using Guided Preemption vs Blind Preemption as used in AUTOVON. The reduction was less for Blind-Back Preemption.

Results obtained with Remote Earth-Station Querying were similar to those obtained previously with a Steady-State Analysis program. Mixed-Media Routing with Remote Earth-Station Querying performed similarly to Adaptive Mixed-Media Routing when the satellite was destroyed in network DSN1. When additional damage occurred in DSN1, performance was intermediate between Adaptive Mixed-Media Routing and Mixed-Media Routing.

2.5 INVESTIGATION OF MACHINE INTELLIGENCE TECHNIQUES FOR SYSTEM CONTROL

A study was undertaken during FY85 of the potential applicability of Machine Intelligence techniques to enhance the effectiveness of System Control in the military communications environment. The progress of the study is described below, along with the conclusions. A number of interactions took place during the study with personnel in the Communications Division of the Rome Air Development Center (RADC) who are working on the future development of System Control; and in fact, RADC is sponsoring a program at Lincoln Laboratory in FY86 to address the two main recommendations resulting from the study effort. These recommendations are as follows:

(a) In the near term, apply modern Expert System technology to the practical implementation of an automated aid for Tech Control (as described below) for the worldwide U.S. military network of some 61,000 dedicated circuits; and

(b) Concurrently, pursue a longer term, more research-oriented development of architectures and techniques for application of Machine Intelligence to System Control for the Defense Communications System (DCS), which comprises the entire worldwide structure of both switched and dedicated voice and data circuits.

2.5.1 System Control Characteristics and Problems

The general concept of "System Control" refers primarily to those actions that can be taken by human operators at control centers to counteract events that cause network performance degradation. Tech Control as described below is included within, and is in fact the foundation layer of, System Control. The functions of System Control can be divided into three broad categories, distinguished by the time scale over which they are to take effect:

(a) Long-range network design, planning, and administration;

(b) Near-term response (on the order of hours to days) to changes in traffic patterns and connectivity, based on previously prepared contingency plans; and

(c) Fast adaptive response (on the order of minutes) to sudden events such as major network damage due to disaster or enemy action.

36

A hierarchical worldwide structure exists for military System Control, originating at the Defense Communications Agency (DCA) in Washington. The first two categories (or time scales) of System Control listed above are readily accommodated in the current administrative and technological environment, but the third category is not so clear-cut. Rapid and effective response to severe network degradation depends upon having a highly skilled decision-making entity with the capabilities of (a) rapidly gathering reliable information about the new situation from all relevant points in the network; (b) correctly diagnosing the precise nature of the problems and their causes; (c) correctly identifying the best available strategy for alternate routing, repair, etc. to optimally respond to the damage; and (d) reliably implementing the chosen strategy throughout the network. The situation is further complicated in that the decision-making capability should be distributed, in the interest of survivability, yet this distributed structure should be able to function as effectively (even after major parts of it are destroyed) as if it were a centralized authority with omniscient knowledge of all remaining assets.

Because of the daunting nature of the problems in achieving these sophisticated capabilities, current practice of the third category of System Control is rather rudimentary and limited in effect. The most advanced facilities presently in existence for switched telecommunications network control, for example, are arguably those of AT&T. Their international Network Operations Center was visited during the study, at Bedminster, New Jersey, and their problems were examined. Their worst difficulty is the provision of round-the-clock shifts of highly skilled human Network Managers and assistants in the hierarchy of control centers. The network test and status measurement tools available to these managers are limited essentially to coarse-granularity measures of trunk and facility congestion. They have assorted heuristic remedies available, such as blocking calls close to the source if they are bound for a highly congested destination area, or substituting alternate long-haul routes around congested tandem-switching nodes. Some of the heuristic algorithms function automatically, but in all unusual situations the remedies are implemented by a chain of verbal orders ending at local switching centers, where technicians activate the required software changes in switch data bases.

Changes and improvements are being vigorously pursued by AT&T in fast-response network control capability, and some of the philosophy of these changes is applicable to the DCS. Work is in progress to apply Expert System technology to alleviate the chronic shortage of skilled network managers. New generations of telephone switches, with more advanced status reporting and flexible electronic control capabilities, are being installed. Plans are being developed to modernize and automate the control centers, and Bell Laboratories researchers outlined these plans for Lincoln representatives during the Bedminster visit.

The most obvious distinction between the System Control objectives for military and civilian switched telephony systems is the intended beneficiary of service maintenance and restoral. The Bell System must seek to maximize revenue, by guaranteeing the best achievable level of service to all customers. The DCS, in contrast, is driven by a multilevel precedence structure which is entirely willing to sacrifice service for less important users in the interest of guaranteeing connectivity for the highest command levels. It has been said that the Bell System is designed for

Mother's Day, while the military system is designed for D-Day. In fact the DCS System Control structure as it exists today depends primarily on its built-in features and options to meet its objectives; the system has little adaptive capability thus far to respond to unexpected network changes, other than issuing command messages to implement prepared contingency plans. Both the Bell System and the DCS have centralized hierarchical control structures, with limited alternate-site backup for key command centers, and are correspondingly deficient in survivability.

2.5.2 Technical Control Characteristics and Problems

In the course of the FY85 study it became clear that a problem of major importance for military communications is maintenance and restoral of the worldwide network of full-time dedicated voice and data circuits for critical users, and that the control centers charged with this mission can greatly benefit by automation. These centers are known as Technical Control facilities, and there are some dozens of them around the world servicing a network of some 61,000 circuits. A typical Tech Control center deals with one or two thousand circuits which come into the building and appear on manual or electronic patching facilities, and are cross-connected to outgoing circuits which continue on to the next Tech Control center or to an end user. These circuits employ all available types of transmission media (e.g., land lines, military and commercial satellite hops, fiber optics, HF radio), and span broad ranges of digital bit rates, analog bandwidths and circuit quality requirements.

Tech Control functions include routine circuit test and maintenance; fault isolation and diagnosis; and setting up and taking down circuit connections via patch panels, in response to standing orders and new requests, including the option of restoring service for higher-precedence users by seizing and reallocating circuits of lower precedence. These Tech Control operations can be very complex, because of the profusion of circuit media, differences among equipment from various manufacturers, and age and degree of sophistication of facilities. In addition, it is quite possible for multiple/simultaneous problems to create an information input rate so high that even skilled operators can find it impossible to cope effectively.

During the study a number of operating Tech Control centers were visited, and senior Tech Controllers were interviewed at length. The list of facilities visited included a spectrum of types and missions in each of the military departments, as follows:

- 2045th Information Systems Group (Air Force), Andrews AFB, MD
- Pentagon Telecommunications Center (Army), Washington, DC
- NAVCAMS/LANT (Navy), Norfolk, VA
- USAISC-ECTC (Army), Ft. Detrick, MD
- 1999th Communications Squadron (Air Force), Sunnyvale AFS, CA

A picture gradually emerged of a highly skilled and disciplined military specialty area plagued by many problems, the worst of which is gradual erosion of the base of trained personnel. High-paying civilian jobs tend to lure even the more junior Tech Controllers away from the

military, and the career specialists are gradually reaching retirement age. Government funding pressures tend to cause Tech Control centers to be typically undermanned, and replacements who do come in are likely to be new graduates of service schools who know only the barest rudiments of their field, requiring lengthy additional training before they can be of significant help.

The concept of an Expert Systems-based aide for Tech Control was discussed with the senior Controllers, and they responded very positively. The notion of capturing the expertise of human experts in a form that makes it readily accessible to less-skilled operators was well received. In particular, a highly attractive feature of the postulated system was the possibility of using it as a training aid for the constant flow of unskilled new personnel; it appears that such training is a major workload for senior Controllers.

2.5.3 Recommendations for System Control Evolution

As noted above, the conclusion of the study was a two-part recommendation for future work on improvement of System Control: a near-term application of Expert System technology to Tech Control center automation, and a concurrent longer term research program to develop Machine Intelligence architectures and techniques for worldwide control of the Defense Communications System.

The operation of a Tech Control center is a highly knowledge-intensive process which has many features in common with successful Expert System applications in recent years. It was found to be straightforward (although unquestionably painstaking and detailed) to extract explanations from the Tech Controllers of the processes they go through in both routine and emergency operations. Figure 18 illustrates a top-level structure for a software system incorporating such knowledge. The communications network at the right in the figure is the target environment, and it includes facilities both within the Tech Control center and in the external environment. For the initial stages of the system development it is assumed that human operators provide the coupling between the Expert System and the facilities being sensed and controlled; electronic sensing and actuation devices to provide such coupling are readily available and can be added when appropriate.

The human operator at the upper left in the figure is the Shift Supervisor responsible for the decisions and actions taken. With the aid of the Performance Evaluation Module, he evaluates network status and fault conditions as indicated by preprocessed information provided by the Signal Interpretation Module, and selects corrective actions to restore satisfactory network performance. The Planning Module translates these decisions into specific orders for actions to be taken in the target environment. All three modules draw upon their knowledge bases, which have been extracted from expert human controllers in the course of the system development. All three of the Modules make use of the System Simulation, which is a representation in software of every circuit, connection, and relevant characteristic of the target environment. In particular, the Trainer can exploit the Simulation by injecting selected system operation and trouble situations into it, whereupon it generates a set of displays and alarms and presents them to the Trainee; the latter then uses the Expert Tech Controller System, by means of the operator I/O ports, to solve

39
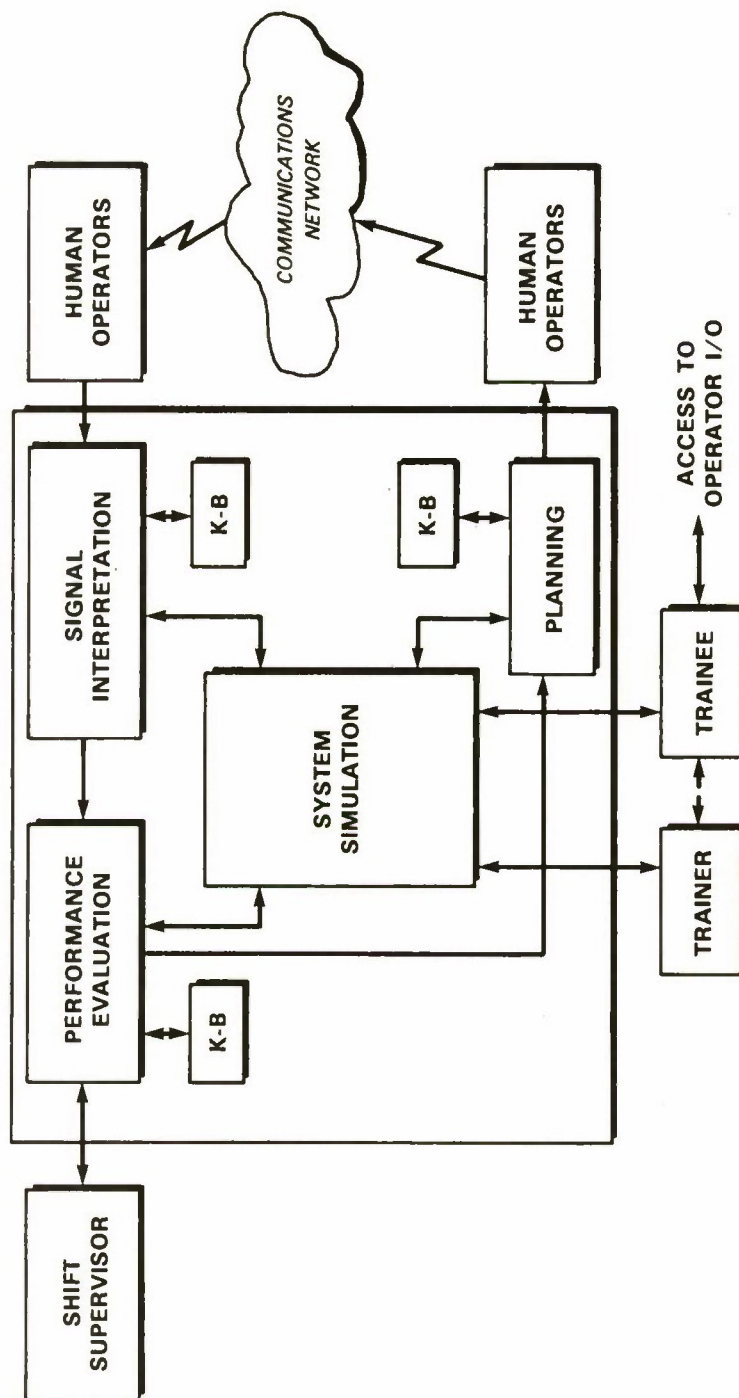
Figure 18. *Structural elements of an Expert System for Tech Control.*

74074-18

the problems set for him by the Trainer. The Trainer's role in the training process is thereby reduced to occasional checking and supervision, in sharp contrast to the detailed step-by-step instruction required of him in the current environment.

The long-term development of Machine Intelligence-Based System Control is seen as a challenging research problem extending over several years. A reasonable approach would be to study and define the architectural requirements for such a system; to define a test bed for development and evaluation of the concepts; and to carry out an extended interactive program of testing of candidate Machine Intelligence strategies for distributed System Control.

3. EISN INSTRUMENTATION AND INTEGRATION

3.1 INTRODUCTION

The FY84 Annual Report[3] described progress in designing, implementing, and integrating the system hardware and software for the EISN Advanced Routing and System Control Test Bed. This test bed was to consist of a number of flexible telecommunications network nodes, geographically distributed across the United States, each incorporating a modern stored-program-controlled telephone switch controlled by a programmable outboard processor (the RCP, or Routing/Control Processor). The RCPs were to be joined by a packet-switched, digital Common-Channel Signaling (CCS) network, and internode trunking was to be provided by a mix-of-transmission media, including land lines and a Demand-Assigned Multiple Access (DAMA) satellite channel. Special interface cabinets would support multilevel preemption and would present standard telephone trunk interfaces to the switches for the satellite and terrestrial transmission channels.

FY85 saw the completion and deployment of these facilities, with certain changes dictated by future program and support considerations, as outlined below. As successive sites were outfitted with their equipment, the experiment sequences described in Section 2 were performed.

3.2 MULTISITE SYSTEM INTEGRATION

The original deployment plan for the EISN Advanced Routing and System Control Test-Bed Network, as illustrated in Figure 8 of the FY84 Annual Report,[3] was as follows:

DEC site	UTX-1200 switch (United Technologies LEXAR)
Lincoln site	UTX-1200
MILDEP sites	
Ft. Huachuca, AZ	UTX-1200
Ft. Monmouth, NJ	SL-1LE (Northern Telecom)
RADC, Griffiss AFB, NY	(switch type undecided as of FY84)

Two factors which came to light during the early part of the year changed the deployment location list, and acted as a driver for the timing. The first was an Army decision to terminate EISN funding for both locations as of the end of FY85, and the second was a DCEC decision not to locate an EISN node on DCEC premises. The immediate consequence of the first factor was a speedup of delivery of EISN equipment to the two Army sites, to permit the greatest possible amount of experimentation by Army site personnel prior to the end of the year. The second factor led to an agreement between DCEC and RADC that both the Lincoln and DCEC UTX-1200 switches would be permanently deployed at RADC where, with a third switch to be procured by RADC, they would form a flexible, programmable 3-node in-house telecommunications

test bed. Concurrently, RADC chose to purchase a Northern Telecom SL-1LE as the nexus of the third node.

During the field installation efforts, each of the EISN sites was outfitted with a Winchester hard disk system for its Packet/Circuit Interface (PCI) equipment already in place. This corrected what had been a troublesome nuisance in operating the PCI equipment: it was now possible to reboot the system in about one minute, rather than the 15 min. previously required with the TU58 cassette tape drives originally delivered with the PCIs.

3.2.1 Site Equipment and Facilities

The timetable for deliveries of RCPs and associated facilities to the sites was as follows:

Ft. Huachuca	November 1984
Ft. Monmouth	January 1985
RADC (2 UTX-1200 nodes)	August-September 1985
RADC (SL-1LE node)	January 1986

An RCP and its special interfaces were shipped from Lincoln to Ft. Huachuca in late October 1984. A team of Lincoln staff and contractor personnel spent the week of 5 to 9 November at Ft. Huachuca wiring and checking out the equipment, integrating it with the existing UTX-1200 switch and PCI facilities there, and demonstrating for site personnel the basic capabilities that were available at the time of installation. These included (a) establishment of local calls on the UTX-1200 phones under RCP control; (b) setup and execution of emulated traffic experiments within the RCP, together with statistics collection and reporting; (c) long-distance calls between Lincoln and Ft. Huachuca UTX-1200s via dial-and-hold trunking, under RCP control, with signaling carried by CCS; and (d) satellite channel calling (without RCP involvement) between UTX-1200 phones at Ft. Huachuca and TOE phones at Lincoln. Actual completion of the fourth item was delayed due to certain data-base problems within the Ft. Huachuca UTX-1200, which were subsequently corrected.

The attendant console interface development for the SL-1LE switch, under subcontract to Northern Telecom, Inc. (NTI) as reported in the FY84 Annual Report,[3] was completed in December 1984. Modem and switch interface assemblies were loaned to NTI by Lincoln, and debugging of the attendant console interface was carried out by means of long-distance modem connections linking an RCP at Lincoln with the interface equipment and a test-bed SL-1 switch at NTI's laboratory in Santa Clara, California. After completion of these tests the RCP and associated interface cabinet were shipped to Ft. Monmouth from Lincoln, and the prototype SL-1 Attendant Console Interface Assembly was shipped to Ft. Monmouth by NTI. A team of Lincoln staff and contractor personnel, together with the NTI design engineer responsible for the SL-1 interface, completed the site integration of this equipment in January. Certain of the basic capabilities enumerated in the previous paragraph were demonstrated immediately, and the

remainder followed the correction of certain switch data base and interface bugs. The NTI engineer returned home and proceeded with the construction of three delivery copies of the SL-1 attendant console interface, as specified in the contract; subsequently, two of these were delivered to Ft. Monmouth, and one was sent to Lincoln.

The RADC installation process was planned with considerable care, since it was to be much more complex than the Army site operations, with two RCPs, two switches, and two interface cabinets to be completely wired and checked out. A preliminary planning and site survey trip was made to RADC in February, to identify particular problem areas that should be resolved in advance of the actual installation, such as equipment layout and on-site support requirements. One example of an issue that came to light at this time was that the on-base telephone plant at RADC can accept only pulse dialing on subscriber lines; this required that Lincoln modify the software in the Central Office Controller equipment, which had previously accommodated tone dialing only. Shipment of the two UTX-1200 switches and their RCPs and interfaces to RADC took place in late July 1985, and the installation process required several visits by Lincoln teams. The culmination of this installation effort was the system introduction and equipment demonstration meeting conducted by Lincoln personnel at RADC on 26 September, as described in Section 4.

The SL-1LE ordered by RADC for the third in-house EISN node was delivered and installed at RADC in November. By prearrangement between RADC and Army personnel, one of the SL-1 Attendant Console Interface Assemblies (viz., the one at Lincoln Laboratory) was provided to RADC for use with the new switch. The pacing item on installation and integration of the third node at RADC is certain components of the special interface cabinet, which had been funded by RADC for FY86 procurement and were due to be delivered to Lincoln about 1 January 1986. The plan is to complete the site installation as early as possible in CY 1986, after delivery and checkout of the required components. At that time the 3-node network at RADC will be exercised by means of a series of real and phantom calling experiments.

3.2.2 Experiment Setup and Initialization Facilities

Automated dialing of terrestrial speech and signaling paths between RCP nodes was introduced this year, in the form of the "connect" utility program, to lessen drudgery and errors during the setup of RCP experiments. Experiments involving RCPs at more than one site require telephone connections dialed between the sites to support terrestrial traffic, both CCS signaling and speech. The signaling links pass through modems, and are dialed by passing control characters to an autodialer that services a bank of the modems. Speech connections are dialed by passing commands to the CO trunk controllers that act as interfaces between the trunk ports of the EISN switch and the intersite telephone network. Dialing these connections by hand, i.e., looking up phone numbers and typing device specifications and digits to the autodialer or the CO controllers, can be tedious and error-prone. In addition, redialing is frequently necessary because of congestion at one of the sites. Experiments proceed more smoothly if dialing can be automated.

The "Connect" program has been written to ease intersite dialing by providing the following features.

    (a)   Dialing and answering by mnemonic link names instead of phone numbers.

    (b)   Dialing and answering by the names of networks (lists of links).

    (c)   Hanging up by names of links or networks.

    (d)   Display of predefined links and networks.

    (e)   Reports of the connection progress.

    (f)   Access to the primitive commands of the autodialer and CO controllers.

When invoked, "Connect" operates interactively, accepting the user's commands from displayed menus. The user starts in the top menu, whose prompt is "Connect:". The following kinds of commands are available.

    (a)   Servicing networks.

    (b)   Servicing individual links.

    (c)   Displaying predefined networks and links.

    (d)   Invoking other menus, which access primitive commands to control devices.

    (e)   Displaying the top menu itself.

The following is a list of representative command examples:

    (a)   "nc RADCRA" causes the list RADCRA of network commands to be executed. Normally a companion command will be issued at each of one or more sites to complete the connection process. For instance, the network RADCRA at radcrcp may have a command to call the ccs link to arcp, while the network ARLL at arcp would have a command to answer the ccs link to radcrcp. (Since modems auto-answer, "answer" really means "verify answering" in this case.)

    (b)   "lh accs" causes the hanging up of the link named "accs".

    (c)   "n" displays a list of the names of all the predefined networks.

    (d)   "c" invokes the primitive CO device handler.

    (e)   "v" lets the user view the top menu, as follows:

a	run autodialer from terminal
c	run CO controller from terminal
e	run E&M controller from terminal
l	display choice of links
la <link>	answer a link. Ex: la accs

lc \<link\>	call up a link. Ex: lc accs
lh \<link\>	hang up a link. Ex: lh accs
n	display choice of networks
nc \<net\>	connect a network Ex: nc LLRA
nh \<net\>	force hangup of a network. Ex: nh LLRA
nk \<net\>	kill net (allows new net cmd). Ex: nk LLRA
s	status of all connections
t	trace toggle
v	print this menu

quit with CTRL-c

The autodialer primitive menu is as follows:

a	abort dialing
d \<digits\>	dial to typed phone number
h	hang up the phone
m \<modem\>	choose a modem
q	quit to top-level menu
t s or c	set or clear trace mode
v	print commands

The CO Controller primitive menu is as follows:

0	command e lead low.
1	command e lead high.
2	command onhook.
3	command offhook.
4	command disable tone-detect reports.
5	command enable tone-detect reports.
c	display controlled chassis and trunk
c \<chasid\>	choose chassis Ex: c 2
n	command preemption tone off.
p	command preemption tone on.

47

q	quit to top-level menu
r	command reset.
d <digits>	command dial. Ex: d 8635500
s	command status.
t	display controlled chassis and trunk
t <trunkid>	choose trunk. Ex: t A
v	print commands.

3.3 RCP SOFTWARE DEVELOPMENT

During FY85 the RCP software needed to support the experiments and demonstrations described elsewhere in this report was completed and checked out. The implementation adhered to the overall design described in the previous Annual Report[3] with some minor changes based on experience and some extensions to provide new features. In addition to correcting system problems as they were discovered, software development activities included the following:

(a) Creation of a new module to support the SL-1 Switch Interface.

(b) Development of a modified CCS protocol to support the multi-address capabilities of WB SATNET trunking and a program module to handle that protocol.

(c) Changes to handle signaling with the PCI that provides access to the satellite trunks.

(d) Extension of the routing and preemption software to support alternate routing and guided preemption.

(e) Additions to the call sequencer to support real-call preemption through the switches and to handle tandem real calls.

(f) Major enhancement of simulator capability to simplify the running of complex phantom-call experiments directly from menus.

(g) A number of small changes to the menus and displays to make the system easier to use.

3.4 MULTI-RCP EXPERIMENT CONTROL FACILITIES

A multi-node RCP experiment is one in which several RCPs concurrently communicate with each other in order to set up and terminate calls. The calls can be real calls or phantom calls. The latter can be driven algorithmically by a set of parameters, in a predetermined fashion by a precalculated "CALL" file, or on a single-shot basis. One can change the network represented by the interconnections of the RCPs and/or the characteristics of the calls that are made. The end result, a set of statistics maintained by the RCPs, can help one gain some understanding of the behavior of networks, routing strategies, and related issues.

This section describes the operational issues encountered when running a multi-node RCP experiment and the facilities that were developed at Lincoln to make such runs convenient and practical. These facilities have been used regularly on a production basis to run a variety of multi-node RCP experiments. In particular, these facilities were successfully used in the 9-node phantom-call experiment (described earlier in Section 2) which compared RCP behavior with that of the Call-by-Call Simulator.

3.4.1   Operational Issues

The following are operational issues and limitations encountered in running a multiple RCP experiment. It is these limitations that we circumvented at Lincoln, as described in the subsection following this one.

3.4.1.1   Terminals

If one runs in a mode in which each RCP is invoked and controlled from its own physical terminal, then the number of terminals available is an obvious limitation to the number of RCPs that can be run concurrently. However, such terminal usage is extravagant and not really necessary. As described later, facilities were developed for running a multiple RCP experiment completely controlled from one physical terminal.

3.4.1.2   Simultaneous Execution

For an experiment to be meaningful, the RCPs in a multi-node experiment must run concurrently. Additionally, in many experiments the RCPs must also start simultaneously. Because of the scheduling characteristics of time-sharing systems and the inherent difficulties of synchronizing several computers, one can only approach simultaneous initiation of an experiment in several RCPs but cannot achieve it completely. The solution is described below for the problem of starting several RCPs at the same time.

3.4.1.3   Control

Controlling a multi-node experiment can be cumbersome and error-prone. One must issue the same set of commands to each RCP so that its behavior will be similar to that of its partners. Techniques were developed to dramatically ease this problem.

3.4.2   Lincoln Control Facilities

This subsection describes the control facilities and techniques that were developed at Lincoln to overcome the operational limitations described in the previous subsection. These facilities enable convenient running   and control of several concurrent RCPs as part of a multi-node experiment.

3.4.2.1 Access to Several Computers via "todown"

Todown is a program which enables one to log in and run programs on "downline" computers as if one were at the terminals of these computers. Communication between the "host" computer running todown and the downline computers is via RS-232 lines, with the downline computers and their programs generally not being aware that they are not communicating with physical terminals. Todown was originally developed for use in the wideband speech program and was easily adapted for use here.

Commands are provided in todown for dynamically attaching and releasing downline computers. At any given moment one may select a specific downline computer that one wishes to communicate with and make it the "current" one. When one enters "type-through" mode to this current computer, one is in an environment virtually similar to that of sitting at a physical terminal of the computer. Todown's intermediary activities are transparent as one types-through to the computer and receives responses. Yet, when one wishes one can select another computer as the current one and type-through to it.

There is a large variety of commands in todown. Included are commands that provide information on the current status of todown, enable one to slave output to another physical terminal where "watchers" can observe the happenings on the main terminal, download or upload files from the downline computer, and provide other functions.

In the context of multi-node RCP experiments, todown was regularly used to log in from a host computer to several downline RCP computers to invoke and run RCP processes. Heavy use was made of the commands to engage in dialogs with downline computers via "converse" files, and to broadcast character strings to all the attached computers. These facilities are described later in this subsection.

3.4.2.2 Running of Several RCPs in One Computer via "mpx"

The mpx program enables one to control, from one terminal, several independent programs simultaneously running in the same computer. mpx evolved from todown (described above); its development was motivated by the desire to run several logical RCPs in one computer. Because of this evolution, mpx has many of the same kinds of facilities that todown has, as described above.

Communication between mpx and the object programs is via UNIX/VENIX pipes, which are transparent to most user programs. As in todown, one can select a current program for type-through purposes, use converse files for dialogs, broadcast character strings, etc.

Using mpx for multiple RCP experiments is straightforward. One invokes mpx and commands it to "fork" several "Shell" programs. Then one types-through to each Shell and issues the command to invoke a program, in this case an RCP. Later, one can type-through to any of these RCPs in order to issue commands to the RCP or to observe results.

3.4.2.3 Combined Use of "todown" and "mpx"

At Lincoln, todown and mpx were combined to run and control a multi-computer, multi-node RCP experiment. Todown was run to attach and log in on several computers. On each computer mpx was invoked. Each mpx was commanded to fork several Shells, in each of which an RCP was invoked. The result was a multi-node experiment completely controlled from one physical terminal. This was the mechanism used to run the 9-node phantom-call experiment at Lincoln, as described in Section 2.

3.4.2.4 Using Converse Files to Control Operation

Converse files in todown and mpx enable one to specify in a file an expected type-through dialog with a downline computer (todown) or with a program (mpx). The dialog specifies what is typed and what is expected to be typed back. This ensures consistent, error-free typing to a program and provides automated checking that the responses are correct.

In order to control multiple RCPs, converse files may be used either on the todown level or the mpx level. The multi-node experiments at Lincoln used converse files at the todown level. There were two classes of such files: those which established dialogs with the downline computers themselves, and those which engaged in dialogs with the logical RCPs indirectly through mpx. Files were used to log in on the downline computers, to invoke mpx on each computer, to invoke and start all the logical RCPs, to check that all the links between the RCPs were up, to issue sets of control commands to the RCPs and, finally, to start the phantom-call simulation.

3.4.2.5 Broadcasting Commands

In order to issue commands to all the mpx programs and/or all the logical RCPs, commands were implemented in todown and mpx for broadcasting a character string to all the downline computers (todown) or to all the object programs (mpx). In this way simulations could be started at the same time in all the logical RCPs (subject to time-sharing limitations). This facility also made it convenient to broadcast commands to all the logical RCPs even when simultaneous reactions were not required. This obviated the need to successively type-through to each RCP and repeat the same commands, a cumbersome and error-prone process.

3.5 INTERNETWORK VOICE/DATA INTEGRATION FACILITIES

During FY85 several small changes and additions were made to the hardware and software of the voice/data gateways to the WB SATNET.

Small disk systems with both hard and floppy disks were added to the facilities at Ft. Huachuca, DCEC, Ft. Monmouth, and RADC. These systems and the software to use them provide much more rapid and convenient software loading (the hard disk) and distribution (the floppy).

The originally proposed addressing scheme for the packet portions of the EISN network was implemented. This scheme makes the PCIs and LEXNETs into hidden networks that do not have

network numbers in the Internet environment. For compatibility, the previous *ad hoc* addressing is still supported.

The gateways were modified to create WB SATNET streams only when needed instead of maintaining a small stream at all times and enlarging it as required to accommodate offered traffic. The change results in longer call setup times and seems to increase the probability that setup will fail due to problems in the WB SATNET in creating the new stream, but it reduces the average load on the channel and allows more sites to share the channel when conditions are poor.

The gateway software was extended to support the time-stamp option in the Internet Control Message Protocol. This capability is useful in carrying out network performance measurements.

3.6   EISN SYSTEM DOCUMENTATION

Documentation has been provided with the principal EISN system elements at the time of installation at field sites, in general, throughout the project. This material has normally included complete files of manufacturers' manuals as well as specialized documents describing Lincoln-built hardware and software subsystems.

Special emphasis has been placed during FY85 on the preparation of comprehensive documentation for the RCP and associated subsystems. As a general rule, the level of explanation and detail is such that an engineer at an EISN site would be able to find all the information necessary to maintain and support the system. The RCP documentation package is now complete, and is available in hard copy and on-line at RADC; additional hard copies can be obtained at RADC when desired by printing the files. Also, archival copies of the material are retained at Lincoln both in hard copy and on-line.

The following subsections describe the six categories of RCP documentation, including a one-sentence descriptor of each file.

3.6.1   Introduction

doc.ms   Contains a list of the RCP documents and a brief description of each. You are now reading this file.

softmnt.ms   Contains a list of the software packages supplied as part of the RCP project and information on generating new versions.

3.6.2   General Description

These files provide descriptions of various aspects of the RCP and its protocols.

nplan.me   Describes the number plan for EISN. It contains examples of dialing sequences, including special fields for setting precedence and source routing.

ccs.ms	Describes the RCP CCS protocols.
cpnotes.ms	Provides an overview of the RCP software.
calls.ms	Provides a "call state" description tracing a call's state in the RCP.

3.6.3 File Descriptions

These files describe the organization and contents of several RCP files.

conf.ms	Describes the RCP initialization and configuration file.
config.ms	Describes the RCP configuration file from the point of view of allocating hardware resources.
simload.ms	Describes the Traffic File used for setting up parameters for an RCP simulation run.

3.6.4 Execution

These files provide details on the mechanics of booting VENIX and running phantom- and real-call experiments.

bootrcp.ms	Gives instructions for bringing up and shutting down the VENIX operating system.
menu.ms	Describes the RCP menus and displays which provide the means for controlling and examining the behavior of an RCP.
run.ms	Describes how to set up and run a multiple-RCP phantom-call experiment.
real.ms	Describes how one runs real-call exercises.

3.6.5 Auxiliary Programs

These files describe several auxiliary programs used in the running of RCPs.

todown.ms	Describes the todown program which can be used to control several RCP computers (each running one or more instances of an RCP) from one terminal.
mpx.ms	Describes the mpx program which enables one to run several instances of an RCP in one computer.
connect.ms	Describes the use of the RCP command "connect". The "connect" program is used for testing modems, autodialers, and trunk controllers and for dialing CCS and speech links to remote sites.

3.6.6 Equipment

These files provide tabular information on various aspects of the RCPs.

trunk.me	Set of tables that give information about each trunk at each site. Data include equipment number, access code, route treatment, trunk controller unit, and telephone number.
datapath.ms	Set of tables that give the connections and functions of each of the tty ports of each RCP. Paths are traced, where appropriate, through RS-232 switches, error controllers, modems, and phone numbers.
cciprot.me	Description of protocols between RCP and the Northern Telecom SL-1 Attendant Console Interface.
cotest.me	Describes procedures for testing the central office trunk controllers.
cabinet.ms	Describes the equipment and connections for the RCP/Switch Interface.
radcwire.ms	Detailed wire run list for the distribution frame on the RADC UTX-1200 telephone switch.
awire.ms	Detailed wire run list for the distribution frame on the A UTX-1200 telephone switch.

4. EISN EXPERIMENT PLANNING
AND SYSTEM COORDINATION

An important ongoing Lincoln role throughout the EISN program has been that of system-wide planner and coordinator. There have been four main elements in this activity: (a) system engineering for the Wideband Satellite Network; (b) developing the EISN test-bed facilities; (c) planning and coordinating the installation of these facilities throughout the network; and (d) planning and coordinating experiments on the EISN test bed.

Achieving stable and reliable operation with the Wideband SATNET has been a long and difficult process, involving step-by-step debugging of complex system problems in developmental equipment from multiple vendors. Steady progress was made by the Wideband Task Force beginning in March 1983 through a series of concerted on-site work sessions organized by Lincoln, with skilled representatives of each vendor (Linkabit, BBN, and Western Union) pooling their resources. During FY85 each of the equipment items was upgraded, resulting in marked further improvements in network operation. The Western Union satellite channel was improved by switching to a cleaner transponder and by replacing 5-m antennas with 7-m units; the BBN PSAT hardware and software were replaced by up-to-date, heavily tested BSAT versions; and the Linkabit Earth Station Interfaces were replaced by upgraded equipment compatible with BSATs. Installation and integration of the upgraded equipment were nearly complete at all sites (with the exception of the two Army locations, which no longer participate in EISN) by the end of December 1985.

The completion and deployment of the EISN Advanced Routing and System Control Test-Bed equipment were discussed in Section 3 of this report. Experiment planning and coordination activity has placed primary emphasis on establishment and verification of the full range of EISN experimental capabilities, as detailed in Section 2. In addition, extensive work was done in providing the Army sites with the experiment planning and coordination tools needed to set up and carry out their own work. During the Ft. Monmouth RCP installation process in January 1985, Lincoln presented briefings and instruction to site personnel on setup and operation of the facilities. A comprehensive set of experiment plans was outlined, which related to traffic loading and comparison of blocking probability to erlang loss formulas, anticipating the use of emulated RCP traffic at the two Army sites individually and together. A set of notes on these experiment plans was shipped to Ft. Huachuca as well, and was discussed at length with site personnel. During April, as noted in Section 2, people at the two Army sites set up and carried out a series of experiments without Lincoln involvement.

Experiment planning activities for the future have focused on the RADC 3-node in-house EISN network. RADC personnel intend to use the network for various telecommunications technology development and evaluation purposes, such as testing a Resource Sharing Unit currently being developed to implement cost-efficient statistical sharing of trunking facilities. Consideration also has been given to future use of the RADC network as a test bed for the new Lincoln initiatives in automation of System Control, sponsored by RADC, as described in Section 2.5.

REFERENCES

1. Defense Switched Network Technology and Experiments Program Annual Report, Lincoln Laboratory, M.I.T. (30 September 1982), DTIC AD-A128932.

2. *Ibid.* (30 September 1983), DTIC AD-A140609.

3. *Ibid.* (30 September 1984), DTIC AD-A158039.

4. R.P. Lippmann, H.M. Heggestad, and R. Berger, "Switch Features that Permit an Outboard Processor to Control Routing and Preemption," Technical Report 681, Lincoln Laboratory, M.I.T. (6 July 1984), DTIC AD-A146193.

5. H.M. Heggestad, "The EISN Testbed for Evaluation of Network Routing and Control Techniques," MILCOM '85 Proceedings, Boston, Massachusetts, 20-23 October 1985, pp. 649-654.

6. R.P. Lippmann, "Survivable Routing Procedures for Circuit-Switched Satellite-Terrestrial Networks," ICC'83 Conference Record, pp. 17-22.

7. —————, "Routing Algorithm Development and Evaluation for Circuit-Switched Mixed-Media Networks," presented at ORSA/TIMS (Operations Research Society of America/The Institute of Management Sciences) 1984 Annual Meeting, Dallas, Texas, 26-28 November 1984.

8. —————, "New Routing and Preemption Algorithms for Circuit-Switched Mixed-Media Networks," MILCOM '85 Proceedings, Boston, Massachusetts, 20-23 October 1985, pp. 660-666.

9. —————, Technical Report (in preparation).

LINCOLN LABORATORY PUBLICATIONS
ON DCA-SPONSORED PROGRAMS
1975 TO 1985

ANNUAL REPORTS

Speech Evaluation Annual Report, Lincoln Laboratory, M.I.T. (30 June 1975), DTIC AD-A021239.

Speech Evaluation Annual Report, Lincoln Laboratory, M.I.T. (30 September 1976), DTIC AD-A041246.

Network Speech System Implications of Packetized Speech Annual Report, Lincoln Laboratory, M.I.T. (30 September 1976), DTIC AD-A045455.

Network Speech Processing Program Annual Report, Lincoln Laboratory, M.I.T. (30 September 1977), DTIC AD-A053015.

Network Speech Processing Program Annual Report, Lincoln Laboratory, M.I.T. (30 September 1978), DTIC AD-A068318.

Voice Conferencing Technology Final Report, Lincoln Laboratory, M.I.T. (31 March 1979), DTIC AD-A074498.

Network Speech Systems Technology Program Annual Report, Lincoln Laboratory, M.I.T. (30 September 1979), DTIC AD-A086768.

Network Speech Systems Technology Program Annual Report, Lincoln Laboratory, M.I.T. (30 September 1980), DTIC AD-A097998.

Network Speech Systems Technology Program Annual Report, Lincoln Laboratory, M.I.T. (30 September 1981), DTIC AD-A112718.

Defense Switched Network Technology and Experiments Program Annual Report, Lincoln Laboratory, M.I.T. (30 September 1982), DTIC AD-A128932.

Defense Switched Network Technology and Experiments Program Annual Report, Lincoln Laboratory, M.I.T. (30 September 1983), DTIC AD-A140609.

Defense Switched Network Technology and Experiments Program Annual Report, Lincoln Laboratory, M.I.T. (30 September 1984), DTIC AD-A158039.

EXPERIMENTAL INTEGRATED SWITCHED NETWORK (EISN)

C.J. Weinstein and H.M. Heggestad, "Multiplexing of Packet Speech on an Experimental Wideband Satellite Network," Proc. AIAA 9th Commun. Satellite Syst. Conf., San Diego, California, March 1982.

H.M. Heggestad and C.J. Weinstein, "Voice and Data Communication Experiments on a Wideband Satellite/Terrestrial Internetwork System," ICC'83 Conference Record, pp. 1-8.

C.J. Weinstein, "The Experimental Integrated Switched Network — A Systems-Level Network Test Facility," MILCOM '83 Proceedings, pp. 449-456.

H.M. Heggestad, "Voice and Data Communications on an Experimental Wideband Internetwork System," J. Telecommun. Networks **3**, 131-146 (1984).

H.M. Heggestad, "The EISN Testbed for Evaluation of Network Routing and Control Techniques," MILCOM '85 Proceedings, Boston, Massachusetts, 20-23 October 1985.

ROUTING

H.M. Heggestad, "A Flexible Routing Algorithm for the Future Defense Switched Network," in Proc. 1980 Natl. Telecommun. Conference, November 1980, pp. 74.3.1-74.3.5.

R.P. Lippmann, "Routing in Packet-Switched Networks: An Annotated Bibliography," Technical Report 585, Lincoln Laboratory, M.I.T. (14 September 1981), DTIC AD-A106155.

R.P. Lippmann, "Steady State Performance of Survivable Routing Procedures for Circuit-Switched Mixed-Media Networks," Technical Report 633, Lincoln Laboratory, M.I.T. (29 December 1982), DTIC AD-A126213.

R.P. Lippmann, "Survivable Routing Procedures for Circuit-Switched Satellite-Terrestrial Networks," ICC'83 Conference Record, pp. 17-22.

R.P. Lippmann, H.M. Heggestad, and R. Berger, "Switch Features that Permit an Outboard Processor to Control Routing and Preemption," Technical Report 681, Lincoln Laboratory, M.I.T. (6 July 1984), DTIC AD-A146193.

R.P. Lippmann, "New Routing and Preemption Algorithms for Circuit-Switched Mixed-Media Networks," MILCOM '85 Proceedings, Boston, Massachusetts, 20-23 October 1985.

MULTIPLEXING

J.W. Forgie, Annual Report to DCA on Network Speech Implications of Packet Speech (30 September 1976), DDC AD-A045455.

A.G. Nemeth, "Behavior of a Link in a PVC Network," Technical Note 1976-45, Lincoln Laboratory, M.I.T. (7 December 1976), DDC AD-A036370.

J.W. Forgie and A.G. Nemeth, "An Efficient Packetized Voice/Data Network Using Statistical Flow Control," ICC '77 Conference Record (June 1977).

P. Demko, "Measurements of Voice and Data Queue Behavior in a PVC Network Link," Technical Note 1977-37, Lincoln Laboratory, M.I.T. (25 August 1977), DDC AD-047101.

C.J. Weinstein, M.L. Malpass, and M.J. Fischer, "Data Traffic Performance of an Integrated Circuit- and Packet-Switched Multiplex Structure," Technical Note 1978-41, Lincoln Laboratory, M.I.T. (26 October 1978), DDC AD-A065449; also IEEE Trans. Commun. **COM-28**, 873-878 (June 1980).

C.J. Weinstein and E.M. Hofstetter, "The Tradeoff Between Delay and TASI Advantage in a Packetized Speech Multiplexer," IEEE Trans. Commun. **COM-27**, 1716-1720 (November 1979).

W. Kantrowitz, C.J. Weinstein, and E.M. Hofstetter, "Prediction and Buffering of Digital Speech Streams for Improved TASI Performance on a Demand-Assigned Satellite Channel," Technical Note 1979-75, Lincoln Laboratory, M.I.T. (15 November 1979), DTIC AD-A081593.

T. Bially, A.J. McLaughlin, and C.J. Weinstein, "Voice Communication in Integrated Digital Voice and Data Networks," IEEE Trans. Commun. **COM-28**, 1478-1490 (September 1980).

C.J. Weinstein, "Statistical Multiplexing of Voice and Data in Broadcast Satellite Networks," in Proc. Fifth Intl. Conf. on Computer Commun. (October 1980), pp. 811-816.

C.J. Weinstein and J.W. Forgie, "Experience with Speech Communication in Packet Networks," IEEE J. Selec. Areas Commun. **SAC-1**, 963-980 (December 1983).

VOICE CONFERENCING

J.W. Forgie, C.E. Feehrer, and P.L. Weene, Voice Conferencing Technology Final Report, Lincoln Laboratory, M.I.T. (31 March 1979), DTIC AD-A074498.

J.W Forgie, "Voice Conferencing in Packet Networks," ICC'80 Conference Record (June 1980), pp. 21.3.1-21.3.4.

SPEECH PROCESSING

P.E. Blankenship, "LDVT: High Performance Minicomputer for Real-Time Speech Processing," 1975 EASCON Proceedings (September 1975), pp. 214A-214G.

E.M. Hofstetter, P.E. Blankenship, M.L. Malpass, and S. Seneff, "Vocoder Implementations on the Lincoln Digital Voice Terminal," 1975 EASCON Proceedings (September 1975), pp. 32A-32J.

B. Gold, "Digital Speech Networks," Proc. IEEE **65**, 1636-1658 (November 1975).

S. Seneff, "A Real-Time Digital Telephone Simulation on the Lincoln Digital Voice Terminal," Technical Note 1975-65, Lincoln Laboratory, M.I.T. (30 December 1975), DDC AD-A021409.

R.J. McAulay, "Optimum Speech Classification and Its Application to Adaptive Noise Cancellation," Technical Note 1976-39, Lincoln Laboratory, M.I.T. (9 November 1976), DDC AD-A036324.

GLOSSARY

BBN	Bolt, Beranek and Newman
BSAT	Butterfly Satellite Interface Message Processor
CCS	Common-Channel Signaling
CPU	Central Processing Unit
CSC	Computer Sciences Corporation
DAMA	Demand-Assignment Multiple Access
DCA	Defense Communications Agency
DCEC	Defense Communications Engineering Center
DCS	Defense Communications System
DES	Data Encryption Standard
DSN	Defense Switched Network
DTMF	Dual-Tone Multiple-Frequency
EISN	Experimental Integrated Switched Network
ESI	Earth Station Interface
FTP	File-Transfer Protocol
ICCU	Interface Controller/Codec Unit
ICMP	Internet Control Message Protocol
IMP	Interface Message Processor
IP	Internet Protocol
NOC	Network Operations Center
NTI	Northern Telecom, Inc.
PCI	Packet/Circuit Interface
PODA	Priority-Oriented Demand Assignment
PSAT	Pluribus Satellite Interface Message Processor
PVT	Packet Voice Terminal
RADC	Rome Air Development Center
RCP	Routing/Control Processor
RSC	Routing and System Control
RTT	Round-Trip Time
SATNET	Satellite Network
TCP	Transmission Control Protocol
TDMA	Time-Division Multiple Access
TOE	Telephone Office Emulator
WB SATNET	Wideband Satellite Network

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER ESD-TR-86-006	**2. GOVT ACCESSION NO.**	**3. RECIPIENT'S CATALOG NUMBER**
4. TITLE *(and Subtitle)* Defense Switched Network Technology and Experiments Program		**5. TYPE OF REPORT & PERIOD COVERED** Annual Report 1 October 1984 — 30 September 1985
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Clifford J. Weinstein		**8. CONTRACT OR GRANT NUMBER(s)** F19628-85-C-0002
9. PERFORMING ORGANIZATION NAME AND ADDRESS Lincoln Laboratory, M.I.T. P.O. Box 73 Lexington, MA 02173-0073		**10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS** Program Element Nos. 33126K and 33126A
11. CONTROLLING OFFICE NAME AND ADDRESS Defense Communications Agency 8th Street & So. Courthouse Road Arlington, VA 22204		**12. REPORT DATE** 30 September 1985
		13. NUMBER OF PAGES 74
14. MONITORING AGENCY NAME & ADDRESS *(if different from Controlling Office)* Electronic Systems Division Hanscom AFB, MA 01731		**15. SECURITY CLASS.** *(of this Report)* Unclassified
		15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

None

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

routing algorithms	Defense Switched Network	circuit-switched network
Experimental Integrated	mixed-media routing	packet switching
Switched Network	preemption	

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

This report documents work performed during FY85 on the DCA-sponsored Defense Switched Network Technology and Experiments Program. The areas of work reported are: (1) development and evaluation of routing and system control techniuqes potentially applicable in the Defense Switched Network (DSN), including investigation of Machine Intelligence techniques for system control; (2) instrumentation and integration of the Experimental Integrated Switched Network (EISN) test bed facility; and (3) EISN experiment planning and system coordination.

DD **FORM** **1473** EDITION OF 1 NOV 65 IS OBSOLETE
1 Jan 73